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： 实验二：FAT32文件安全删除

专业(班)： 信息安全

学 号：

姓 名：

课程名称：

任课教师：

2023年 12 月 19日

目录

[实验2 1](#_Toc155464537)

[2.1实验名称 1](#_Toc155464538)

[2.2实验内容 1](#_Toc155464539)

[2.3实验步骤 1](#_Toc155464540)

[2.4 实验结果 10](#_Toc155464541)

## 实验2

### 2.1实验名称

* FAT32文件安全删除

### 2.2实验内容

FAT32文件系统数据安全删除

要求：

1）作业的重点是分析FAT32文件系统的数据结构；

2）对于给定的文件，要求能够通过分析FAT表识别出来文件所占的簇链；

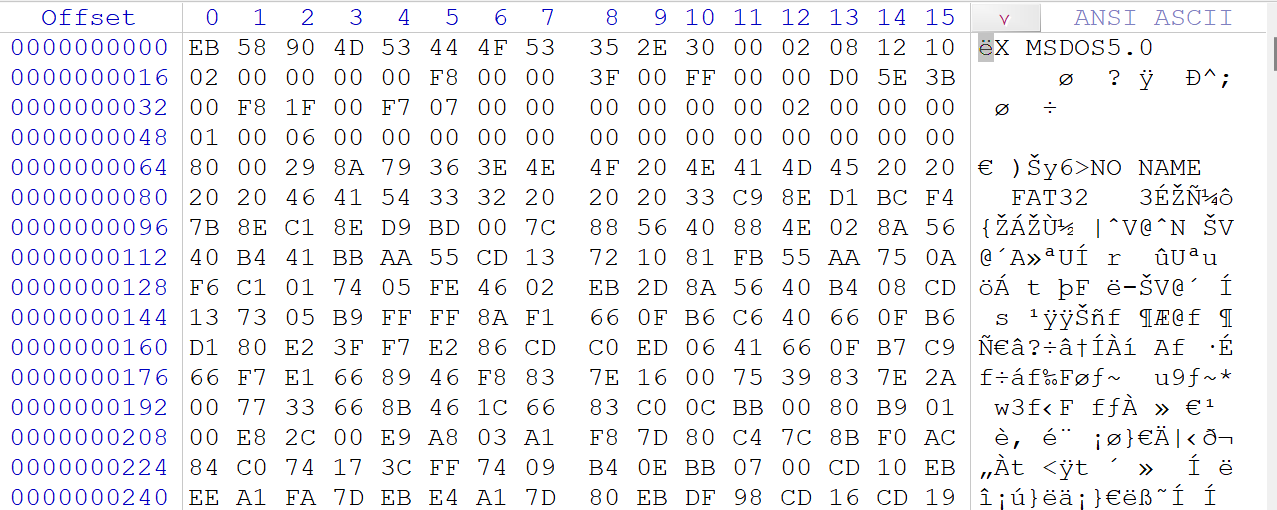
3）对簇的内容重写，以达到安全删除的目的

### 2.3实验步骤



FAT32文件系统整体分布如上图所示

先查看引导扇区



【1】0x00~0x02：3字节，跳转指令。

【2】0x03~0x0A：8字节，文件系统标志和版本号，这里为MSDOC5.0。

【3】0x0B~0x0C：2字节，每扇区字节数，512（0X02 00）。

【4】0x0D~0x0D：1字节，每簇扇区数，8（0x08）。

【5】0x0E~0x0F：2字节，保留扇区数，4114（0x1012），符合FAT1起始地址为4114扇区。



【6】0x10~0x10：1字节，FAT表个数，2。

【7】0x11~0x12：2字节，FAT32必须等于0，FAT12/FAT16为根目录中目录的个数；

【8】0x13~0x14：2字节，FAT32必须等于0，FAT12/FAT16为扇区总数。

【9】0x15~0x15：1字节，哪种存储介质，0xF8标准值，可移动存储介质，常用的 0xF0。

【10】0x16~0x17:2字节，FAT32必须为0，FAT12/FAT16为一个FAT 表所占的扇区数。

【11】0x18~0x19：2字节，每磁道扇区数，只对于有“特殊形状”（由磁头和柱面每 分割为若干磁道）的存储介质有效，63（0x00 3F）。

【12】0x1A~0x1B：2字节，磁头数，只对特殊的介质才有效，255（0x00 FF）。

【13】0x1C~0x1F：4字节，EBR分区之前所隐藏的扇区数，996069376（0x3E5BD0）

【14】0x20~0x23：4字节，文件系统总扇区数，2905104

【15】0x24~0x27：4字节，每个FAT表占用扇区数，2039（0x 00 00 07 E7）。

【16】0x28~0x29：2字节，标记，此域FAT32 特有。

【17】0x2A~0x2B：2字节，FAT32版本号0.0，FAT32特有。

【18】0x2C~0x2F：4字节，根目录所在第一个簇的簇号，2。（虽然在FAT32文件系统下，根目录可以存放在数据区的任何位置，但是通常情况下还是起始于2号簇）

【19】0x30~0x31：2字节，FSINFO（文件系统信息扇区）扇区号1，该扇区为操作系统提供关于空簇总数及下一可用簇的信息。

【20】0x32~0x33：2字节，备份引导扇区的位置。备份引导扇区总是位于文件系统的6号扇区。

【21】0x34~0x3F：12字节，用于以后FAT 扩展使用。

【22】0x40~0x40：1字节，与FAT12/16 的定义相同，只不过两者位于启动扇区不

同的位置而已。

【23】0x41~0x41：1字节，与FAT12/16 的定义相同，只不过两者位于启动扇区不

同的位置而已 。

【24】0x42~0x42：1字节，扩展引导标志，0x29。与FAT12/16 的定义相同，只不过两者位于启动扇区不同的位置而已

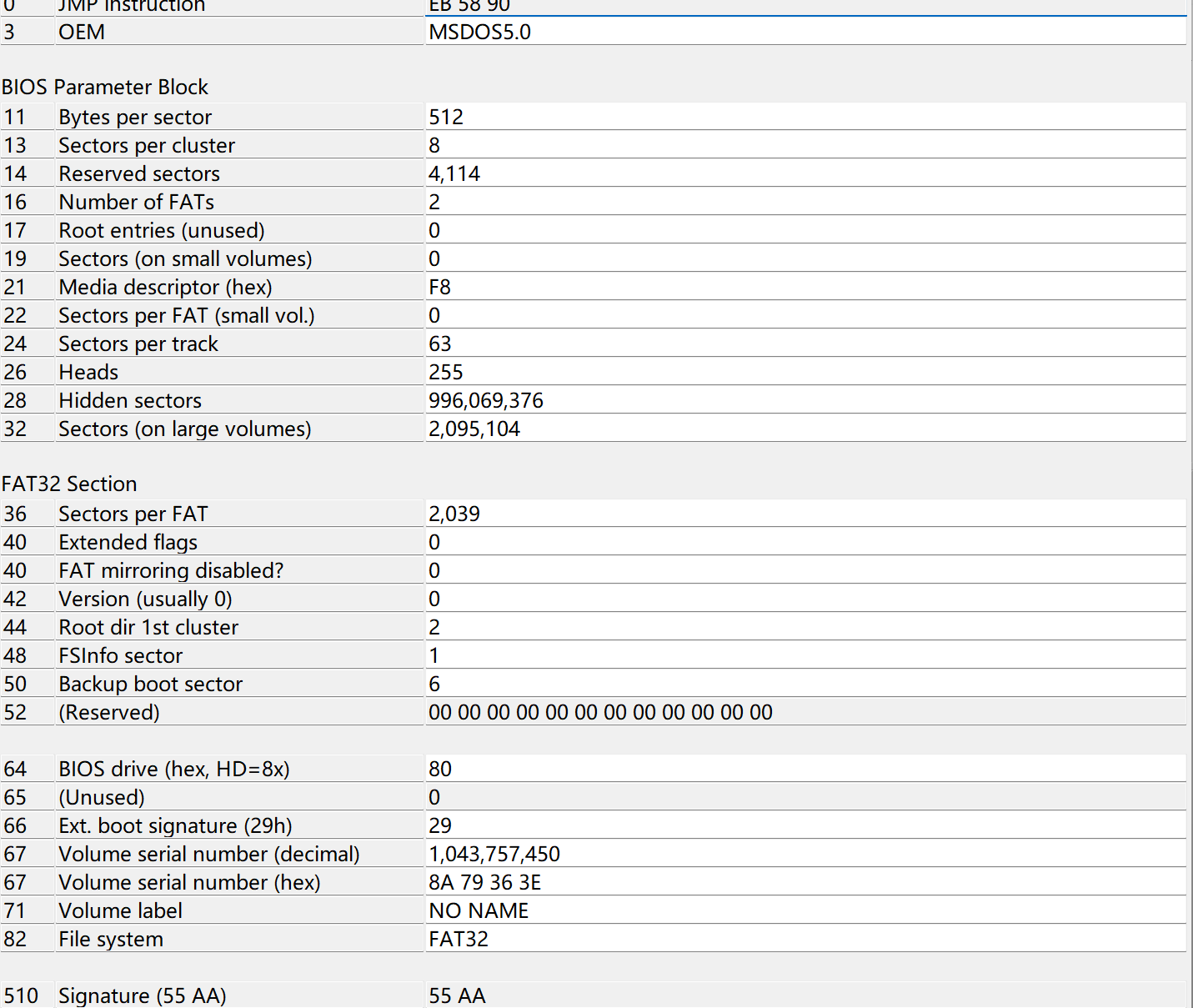
【25】0x43~0x46：4字节，卷序列号。通常为一个随机值。

【26】0x47~0x51：11字节，卷标（ASCII码），如果建立文件系统的时候指定了卷 标，会保存在此。

【27】0x52~0x59：8字节，文件系统格式的ASCII码，FAT32。

★【28】0x5A~0x1FD：90~509共410字节，未使用。该部分没有明确的用途。

【29】0x1FE~0x1FF：签名标志“55 AA”。



查看引导扇区表模板，和我们分析的基本一致

紧接在保留扇区后面是FAT区，其由两个完全相同的FAT（File Allocation Table， 文件分配表）表单组成，FAT文件系统的名字也是因此而来。FAT 表（File Alloacation Table）是一组与数据簇号对应的列表。FAT2紧跟在FAT1之后，它的位置可以通过FAT1的位置加上FAT表的大小扇区数计算出来。

文件系统分配磁盘空间按簇来分配。因此，文件占有磁盘空间时，基本单位不是字节而是簇，即使某个文件只有一个字节，操作系统也会给它分配一个最小单元：即一个簇。对于大文件，需要分配多个簇。同一个文件的数据并不一定完整地存放在磁盘中一个连续地区域内，而往往会分若干段，像链子一样存放。这种存储方式称为文件的链式存储。为了实现文件的链式存储，文件系统必须准确地记录哪些簇已经被文件占用，还必须为每个已经占用的簇指明存储后继的下一个簇的簇号，对于文件的最后一簇，则要指明本簇无后继簇。这些都是由FAT表来保存的，FAT 表对应表项中记录着它所代表的簇的有关信息：诸如是空，是不是坏簇，是否是已经是某个文件的尾簇等。

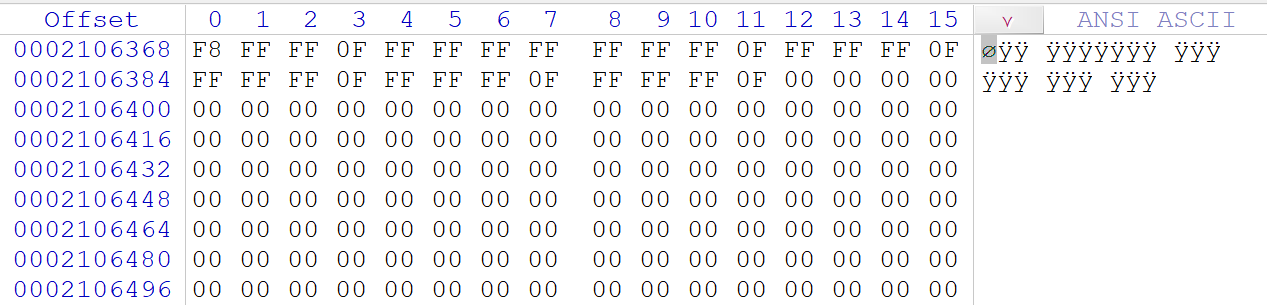
通常情况下，一个FAT把文件系统会有两个FAT表，但有时也允许只有一个FAT表，FAT表的具体个数记录在引导扇区的偏移0x10字节处。上面我们分析了，是2，有两个FAT表，事实也是如此



由于FAT区紧跟在文件系统保留区后，所以FAT1在文件系统中的位置可以通过引导记录中偏移0x0E~0x0F字节处的“保留扇区数”得到，上面我们找到了是4114



那么我们就可以查看FAT表



FAT32中每个簇的簇地址是有32bit（4个字节），FAT表中的所有字节位置以4字节为单位进行划分，并对所有划分后的位置由0进行地址编号。0号地址与1号地址被系统保留并存储特殊标志内容。从2号地址开始，每个地址对应于数据区的簇号，FAT表中的地址编号与数据区中的簇号相同。我们称FAT表中的这些地址为FAT表项，FAT表项中记录的值称为FAT表项值。

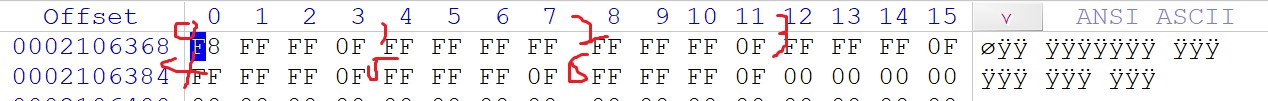
由于簇号起始于2号，所以FAT表项的0号表项与1号表项不与任何簇对应。FAT32的0号表项值总是“F8FFFF0F”。1号表项可能被用于记录脏标志，以说明文件系统没有被正常卸载或者磁盘表面存在错误。不过这个值并不重要。正常情况下1号表项的值为“FFFFFFFF”或“FFFFFF0F”。

如果某个簇未被分配使用，它对应的FAT表项内容为0。

当某个簇已被分配使用，则它对应的FAT表项内的FAT表项值也就是该文件的下一个存储位置的簇号。如果该文件结束于该簇，则在它的FAT表项中记录的是一个文件结束标记，对于FAT32而言，代表文件结束的FAT表项值为0x0FFFFFFF。

如果某个簇存在坏扇区，则整个簇会用0xFFFFFF7标记为坏簇，这个坏簇标记就记录在它所对应的FAT表项中。

在文件系统中新建文件时，如果新建的文件只占用一个簇，为其分配的簇对应的FAT表项将会写入结束标记。如果新建的文件不只占用一个簇，则在其所占用的每个簇对应的FAT表项中新建目录时，只为其分配一个簇的空间，对应的FAT表项中写入结束标记。当目录增大超出一个簇的大小时，将会在空闲空间中继续为其分配一个簇，并在FAT表中为其建立FAT表链以描述它所占用的簇情况。写入为其分配的下一簇的簇号，在最后一个簇对应的FAT表象中写入结束标记。



看上面的图，由于我们是新建的FAT32的磁盘，所以没有什么内容，初始的几个簇只占一个表项，因此都为0x0FFFFFFF

【0号表项】：0x0FFFFFF8;FAT表起始固定标识

【1号表项】：0xFFFFFFFF;不是用，默认值

【2号表项】：0x0FFFFFFF;根目录所在簇，



因为FAT所占扇区为2039，FAT2的起始扇区为4114+2039=6153

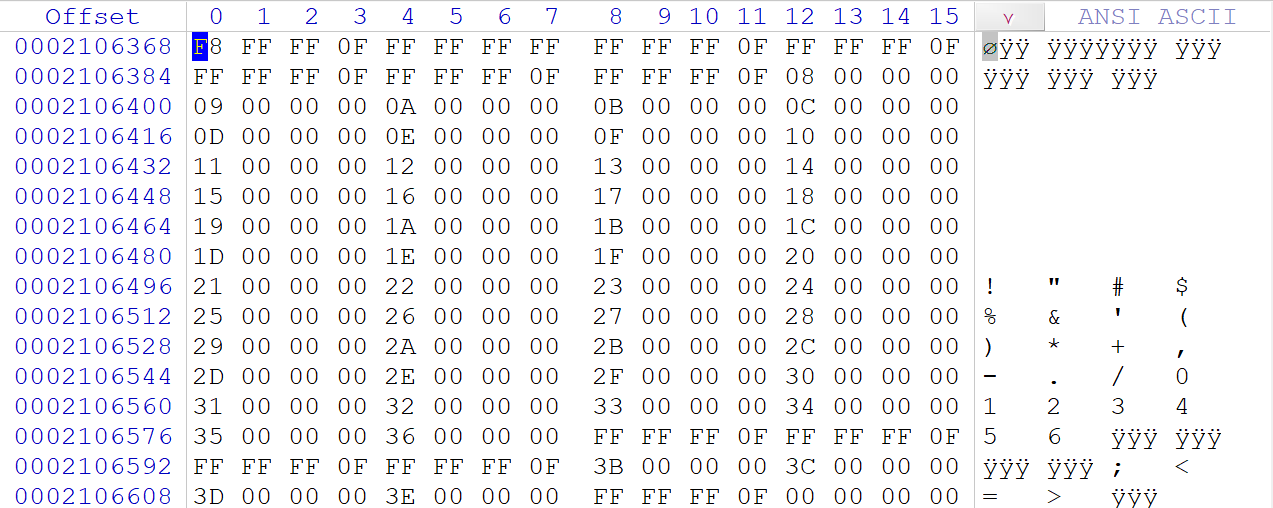
所以根目录的起始扇区在6153+2039 = 8192



可以看到是符合的

为了更好的观察，我新建一个test文件，这里为了更好的观察占多个簇链的情况，我建立一个较大的test文件





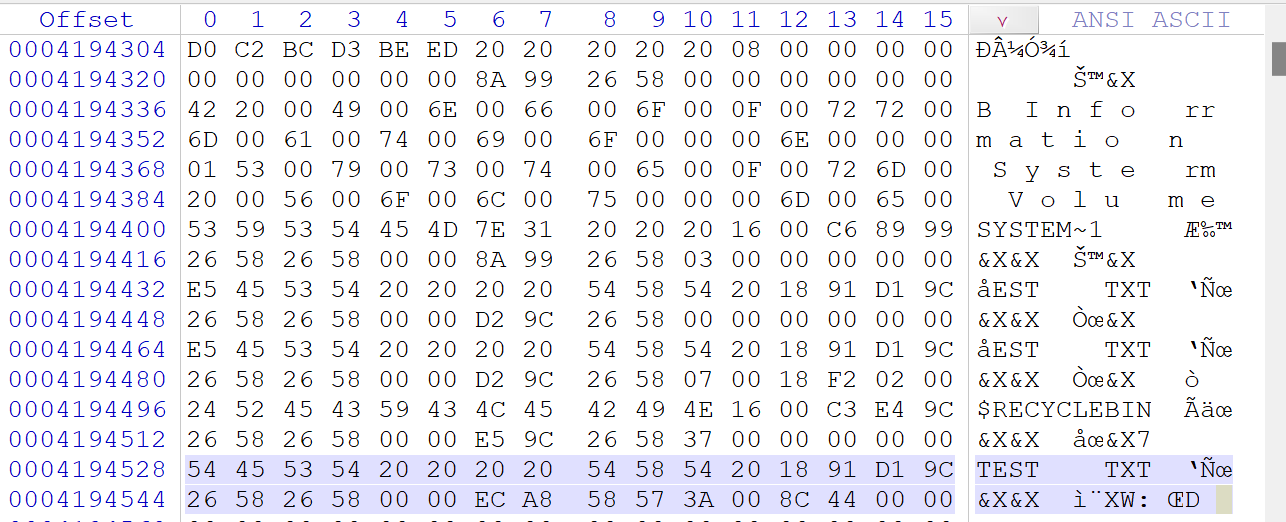
可以看到



我们的test文件起始扇区在8232,正好是我们的第7个表项，8192 + 8 \* 5 = 8232 是正确的

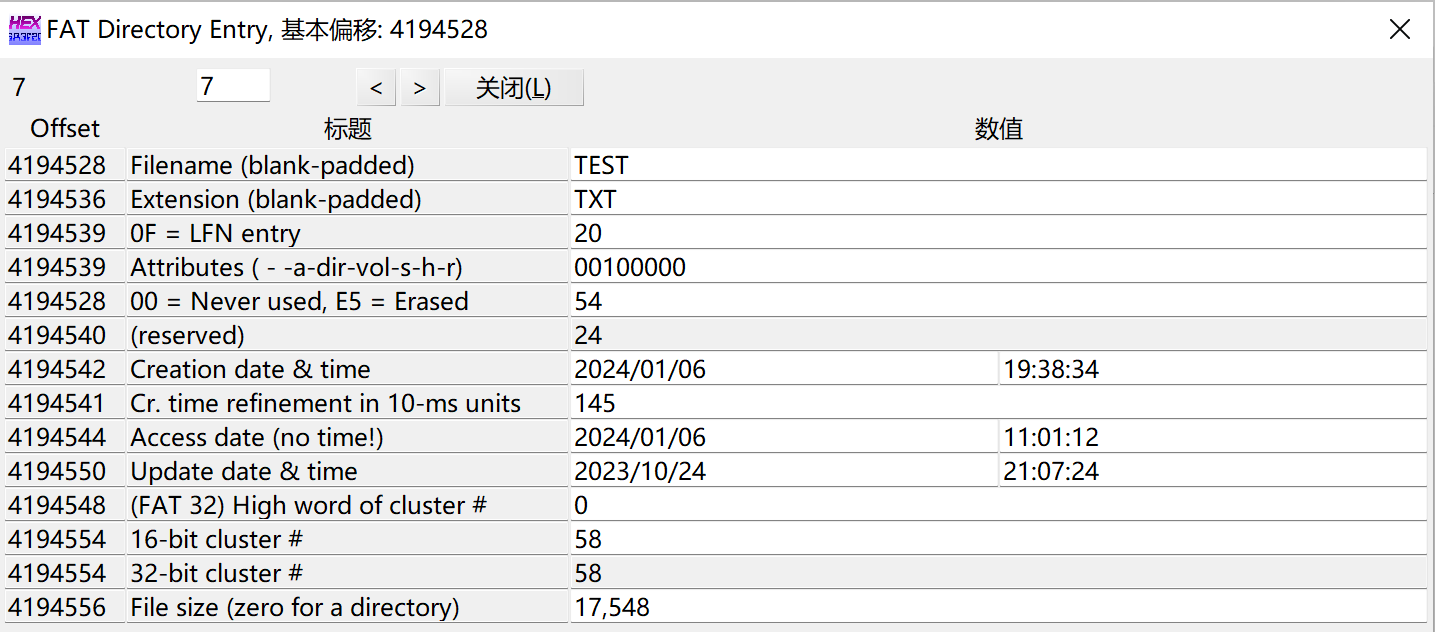
因此7号表项的下一个簇号在9号表项，9号表项的下一簇号在10号表项…到54号表项终止，占了47个簇，大小为47 \* 8 \* 512 = 192512B = 188 KB,和我们查看的文件大小是一致的

查看根目录区，我们上面已经计算得到根目录区的起始扇区为8192，如果想要得到物理地址，则还要加上MBR保留区的大小



因为FAT32目录项占32字节，我们找7号项，找到了我们的TEST文件

我们在根目录区查看7号项



正好是我们的test文件

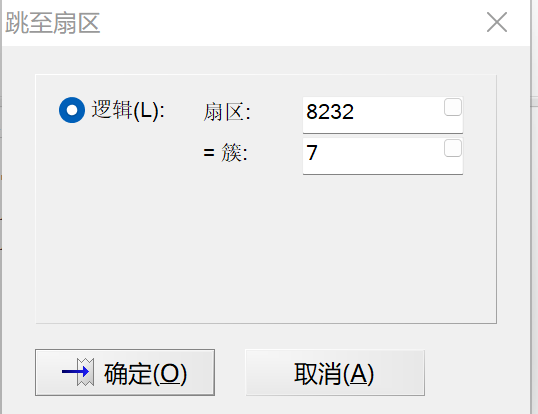
具体每项的定义如下图



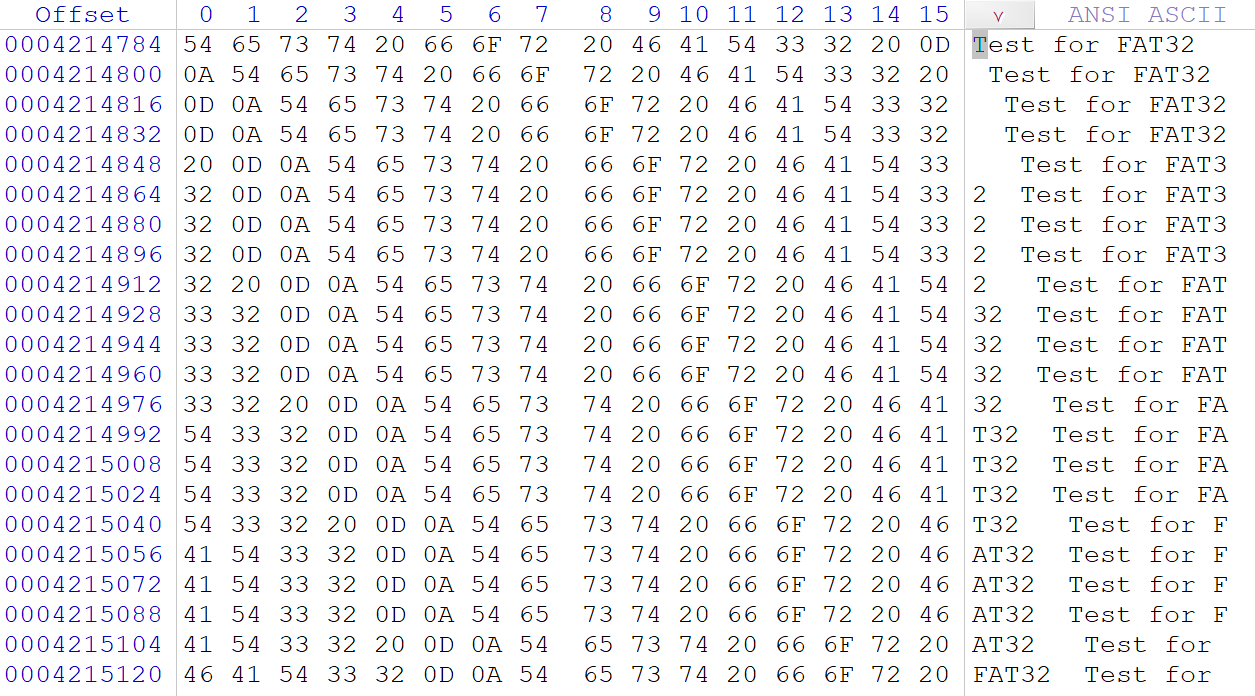
如果是长文件，定义如下：



我们根据簇号跳转至扇区

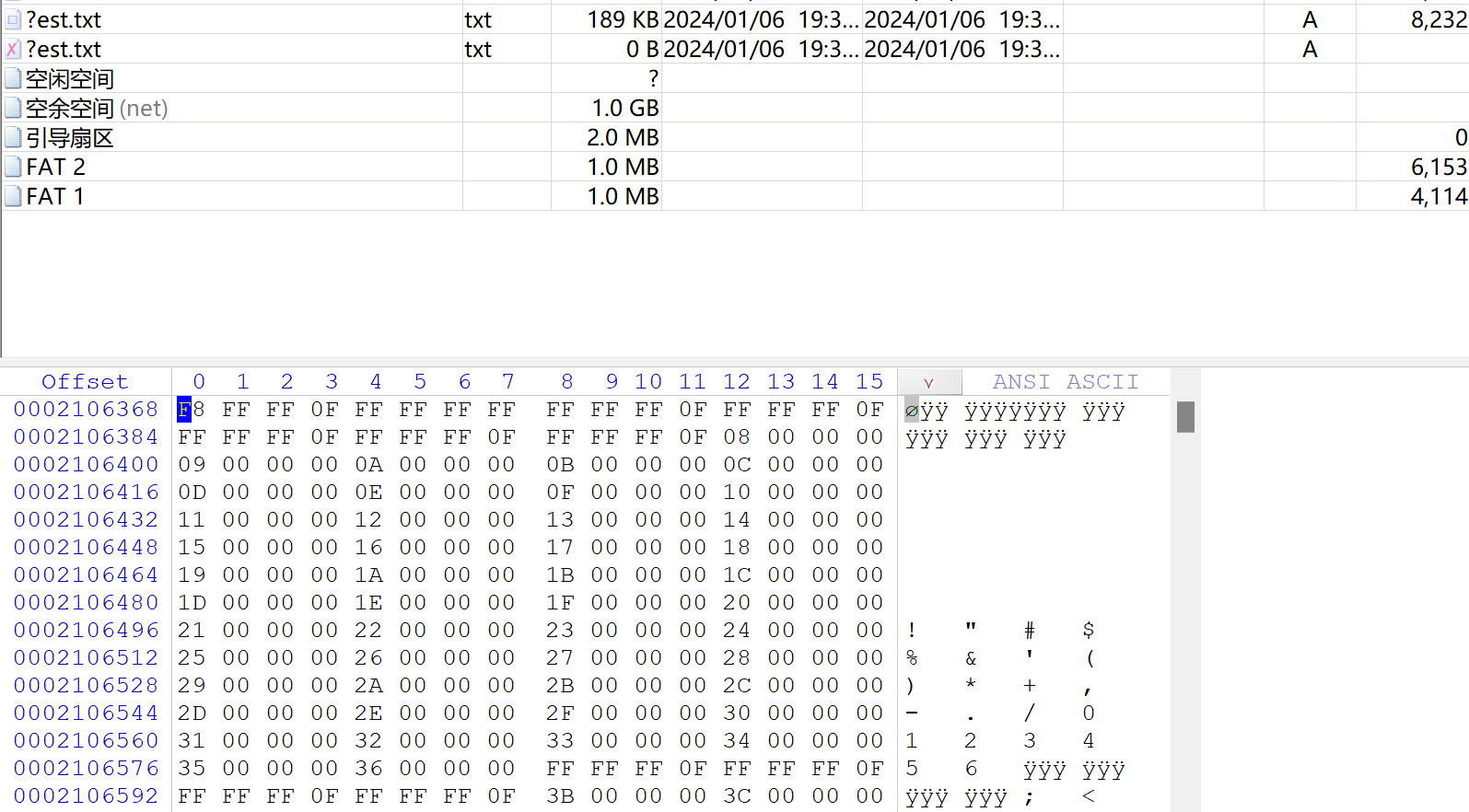


可以看到跳到test文件处



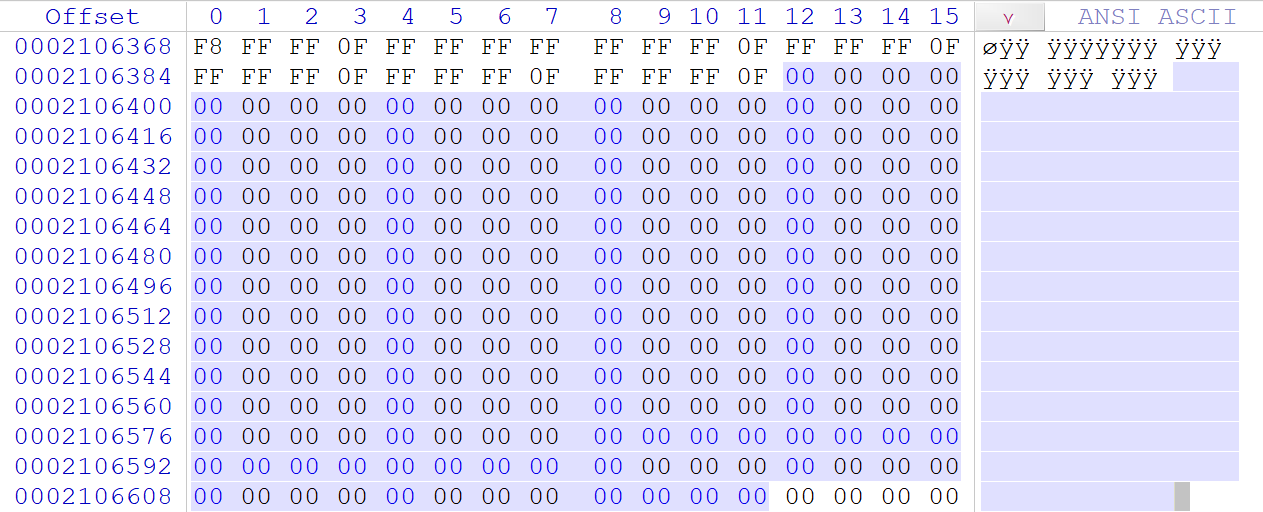
下面要尝试删除文件

我先手动删除了文件，又创建了一个更小的test文件，发现磁盘上还是有删除后的原test的内容

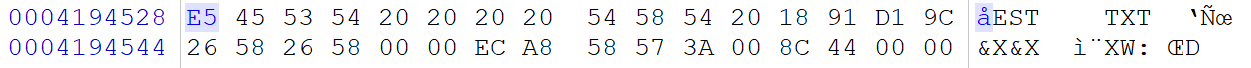


所以del删除文件不会清空簇链表，那么我就把这个老test和新test一起铲除

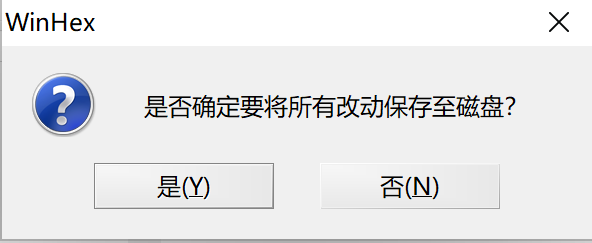
首先我把两个test的簇链表都清空



然后我们跳转至根目录项，把首字节改为E5，表示文件删除，正常来讲，首簇号的高位也应该改成0，由于我的实验磁盘太小，首簇号高位本来就是0，所以就不用改



保存修改



### 2.4 实验结果

执行前



执行后

