第 33 卷 第 23 期 计 算 机 工 程 2007 年 12 月
Vol.33 No.23 Computer Engineering December 2007
·开发研究与设计技术· 文章编号：1000—3428(2007)23—0281—03 文献标识码：A 中图分类号：TP309.2

NTFS 系统存储介质上文件操作痕迹分析

黄步根
(江苏警官学院公安科技系，南京 210012)

摘 要：计算机用户通过文件系统存取数据，文件和文件夹的操作(如增加、删除、修改)会在存储介质上留下痕迹，这些痕迹与文件系统有关。NTFS 文件系统以簇为单位分配和回收外存空间，通过主文件表来进行管理。文章从计算机取证角度探讨 NTFS 文件系统下访问文件(夹)的方法，研究 NTFS 文件系统下文件和文件夹的操作痕迹，并与 FAT 文件系统中的痕迹进行比较。

关键词：数据恢复；计算机取证；痕迹；NTFS；FAT

Analysis of Traces on Storage Media by File Operation for NTFS File System

HUANG Bu-gen
(Department of Forensic Science, Jiangsu Police Institute, Nanjing 210012)

【Abstract】Computer users access data by file system. File and folder operation(such as creation, deletion, and edition) may leave some traces on storage media. These traces are related to file system. NTFS file system allocates and revokes the storage by cluster. It manages by MFT. This paper, from the point of computer forensics, analyzes the method of accessing file for NTFS file system and the traces of the file or folder operating, and compares it with traces of FAT.

【Key words】data recovery; computer forensic; trace; NTFS; FAT

NTFS文件系统已经越来越多地被使用，从计算机取证的角度提取文件操作的痕迹[1]，要保证取证操作的原始性，就不可以调用操作系统提供的文件操作命令，而需要直接存取磁盘扇区，也就需要研究文件系统的相关数据结构；文件和文件夹的增加、删除、修改等操作会在存储介质上留下痕迹，分析这些痕迹，对于信息保密、数据恢复、计算机取证等都具有重要价值。

1 NTFS 卷文件管理

1.1 簇管理

NTFS以簇为基本单位分配和回收存储空间[2]，与FAT结构不同，NTFS卷(volume)从 0 扇区开始划分簇，每簇为 1,2,4 或 8 个扇区，根据分区的大小有一默认值，但是在格式化时可以人工选择。每簇扇区数保存在BOOT扇区(0 扇区)。

NTFS 通过 Bitmap 文件记录所有簇的使用情况，1 个 bit 对应 1 个簇，值为 1 表示已经分配，为 0 表示未分配。FAT 文件系统中的 FAT 不仅标明了数据簇的使用情况，还标明了数据簇的链接关系。

NTFS 使用逻辑簇号(logical cluster number, LCN)和虚拟簇号(virtual cluster number, VCN)来对簇进行定位。LCN 是对卷中所有簇从头到尾进行简单编号，VCN 则是对属于特定文件的簇从头到尾进行编号，以便访问文件中的数据，LCN 是无符号整数，而 VCN 则是带符号整数，VCN 可以映射成 LCN，由 Data Runs 完成这个映射。

NTFS 数据区管理不是采用链接存储，而是采用索引存储，文件通过自己的 Data Runs 建立索引表，一个 Run 就是一个连续存储块，Data Runs 由若干 Run 组成，以 0 结束。每个 Run 包括 3 部分：头，簇数，簇号。簇号用 VCN，第 1 个 VCN 是相对于 0 簇。头占 1 个字节，存放簇数和簇号的字节数(各占 4bit)。对于如下以十六进制数表示的 Run：

31 05 fc b0 12

其中，头 0x31 表示 1 字节簇数(值为随后的 05)和 3 字节簇号(值为 fc b0 12)，即表示 0x12b0fc 簇开始的 0x05 个簇。

假设一个文件的存储分布的 Data Runs 为

31 01 fc b0 12 21 18 bd 49 21 2f 7b a7 00

则分解成 4 个 Run：

31 01 fc b0 12, 21 18 bd 49, 21 2f 7b a7, 00

即文件存储在 3 个连续块：

(1)Run 1: 0x12b0fc 簇开始的 0x01 个簇。

(2)Run 2: 0x12b0fc+0x49bd=0x12fab9 开始的 0x18 个簇，这里将 VCN 0x49bd 换算成 LCN，它相对于前一地址 0x12b0fc。

(3)Run 3: 这里 VCN 是个负数，0xa77b=-0x5885，0x12fab9-0x5885=0x12a234。即文件的第 3 块在 0x12a234 簇开始的 0x2f 个簇。也可以用符号扩展的方法直接相加：0x12fab9+0xffa77b=0x112a234，丢弃最高进位 1，同样得到 0x12a234。

(4)Run 4: 0x00, Data Runs 结束。

文件存储过程中数据簇的分配算法比较复杂，上述文件的存储地址并不完全是依次增加的。

基金项目：江苏省公安厅基金资助项目“基于FAT/NTFS结构的电子证据的发现和提取技术”

作者简介：黄步根(1958－)，男，教授、硕士，主研方向：信息安全，计算机取证

收稿日期：2007-01-25 **E-mail：**bghuang@sina.com

1.2 主文件表

NTFS卷上的任何事物都是文件(为了与平时使用的文件相区别，以下用FILE特指)，FILE通过主文件表(master file table, MFT)来确定其在卷上的位置[3]，每个FILE有固定大小，一般为 1KB。FILE记录了文件的所有数据，每个数据以一个属性来表示，如文件名、文件长度、文件的时间等都是属性，文件的内容也是一个属性，每个属性有一个特征码。属性数据较小时能够存放在FILE记录中，称为驻留的属性，反之为非驻留的属性，通过Data Runs来保存其存储索引表。这一点与FAT文件系统不同，FAT文件系统只在目录区保存了文件的首簇号，还要通过FAT表链接关系才能确定文件的全部存放位置。Data Runs在一个FILE记录存放不下时还可以用扩展属性，增加FILE记录来保存，即一个文件可以有多个FILE记录。

MFT 本身信息记录在 MFT 0(第 1 个 FILE 记录)，取名为$MFT，它存储了整个 MFT 的存储分布，$MFT 的开始簇号在 BOOT 扇区中保存。

1.3 NTFS 卷目录结构

NTFS 中文件名采用 Unicode 编码，直接存储长文件名，转换的 DOS 文件名也以 Unicode 编码，可以选择不保存 DOS 文件名。

NTFS 的目录管理，每个文件和文件夹都有一个或多个 FILE 记录，根目录是 MFT 5。文件夹的内容是该文件夹下的目录数据，记录了各个目录项的属性(文件号，文件名，文件的时间属性等，但不包括文件内容属性)，同层目录采用 B+树结构，按文件(夹)名保持有序，通过文件号指向文件夹内的文件。文件夹的目录项较少时可以直接存储在文件夹的 FILE 记录中，目录项较多时占用数据簇，建立 INDX 记录，存放各目录项的属性。一个 INDX 记录通常占用 8 个扇区(一个或多个簇)，INDX 记录按照 B+树结构组织，一个 INDX 记录是 B+树的一个结点，B+树的“根”结点可以驻留在 FILE 记录中，也可以是一个 INDX 记录。

由于文件名是变长的，因此 NTFS 的目录项是变长的，目录 B+树的调整不是根据目录项数的多少而定，而是根据结点的存放程度而定，一个结点存放不下时要分裂，除根结点外每个结点至少占用 1/2 的存储空间，这与 B+树定义在本质上是一致的。

1.4 NTFS 卷中文件的定位

在计算机上的取证操作，为了保持文件的原始性，不可以调用操作系统提供的文件操作命令，而应根据文件的存储结构直接访问有关扇区。NTFS 中文件定位流程可以归纳为

(1)由 BOOT 扇区得到每簇扇区数并定位 MFT；

(2)由 MFT 5 定位根目录；

(3)对于每个文件，根据文件号在 MFT 中读取 FILE 记录，存取文件内容，驻留的直接存取，非驻留的通过 Data Runs 定位簇号后存取；

(4)对于每个文件夹，小的在 FILE 记录中直接得到其下各个文件(夹)的文件号，大的文件夹根据按文件名顺序组成的 B+树存取到 INDX 记录，进而获得文件夹下文件(夹)的文件号和其他基本目录属性。

2 NTFS 卷文件操作痕迹

计算机取证，首要的问题是掌握什么时间发生了什么事情。有一个重要的文件属性是时间属性，NTFS 中的时间属性有 4 项：文件的创建时间，最后修改时间，最后访问时间以及 MFT 修改时间，全部精确到 100ns。FAT 系统只有前 3 个时间，且精确到 0.2s，最后访问时间只保留年月日。最后访问时间是文件操作痕迹中最敏感的证据特征。文件操作痕迹反映在文件的增加、删除和修改。

2.1 删除文件(夹)的痕迹

删除一个文件(夹)，系统回收其FILE记录，加删除标记，如果该文件(夹)还占用了数据区，系统也回收，在Bitmap中对应位置置 0。回收的FILE记录和数据区可供再分配。FAT系统中的文件删除后，如果文件的存储空间是不连续的，恢复时很难确定文件的链接关系[4]，而NTFS系统中的文件删除后，如果数据空间没有被覆盖，只要FILE记录还存在，Data Runs不会被改变，很容易由它确定文件数据的存储位置，提高了数据恢复的准确性。

删除文件后，该文件所属的文件夹，也要删除该文件的目录项。如果它是非驻留的，将在 INDX 记录内删除它，操作的方式不是加删除标记，而是将该 INDX 记录中后面的数据前移，如果 INDX 空间占用不足一半，将引起 B+树的调整。

文件的删除，在 INDX 记录中可能没有留下有用的痕迹，因为删除的目录项被后面的数据所覆盖，而且由于目录项不定长，定位空闲位置上的目录项比较困难，所以删除文件后的痕迹主要从 FILE 记录获取。不过，如果删除的是文件夹，INDX 记录是可以利用的，数据簇被回收，只要没有再分配，删除文件夹不改变其 INDX 数据簇。FILE 记录空间是非常宝贵的，操作系统删除文件(夹)后，释放出的文件号可能很快就被再分配，数据区则可能要经过很久才被再分配，FILE 记录中保存有文件的存储情况，INDX 记录中则没有，INDX 记录不能定位文件的存储分布，但是 INDX 记录中的目录属性仍然具有证据属性，虽然它不一定能定位和恢复文件内容。

如果文件的内容较少，不需要占用数据区，就驻留在 FILE 记录内，删除文件后，如果 FILE 记录被重新分配了，则文件的内容同时被覆盖。

2.2 增加文件(夹)的痕迹

增加文件，系统首先分配一个 FILE 号，根据文件大小确定是否需要分配数据区，如果分配了，则在 FILE 记录中记录其存储索引。增加文件还在文件所在的文件夹增加目录项，记录文件的文件号和基本目录属性，目录项保存在 FILE 记录或 INDX 记录中，并可能引起所属文件夹 B+树的变化。增加文件夹如同增加文件，只是文件夹的内容就是文件夹下的目录数据，并且根据文件名顺序调整 B+树。

文件改名，由于系统要依据文件名维护 B+树，新位置增加目录，原位置删除目录项，有可能引起 B+树的调整。

2.3 文件(夹)的复制和移动

文件复制，与 FAT 文件系统一样，新位置上所产生的痕迹如同增加文件，文件的目录属性只保留文件的文件名称和修改时间。

文件夹复制，其下文件复制的痕迹同上，文件夹的时间属性全部刷新。

文件移动、跨盘移动时，如同在新盘创建且在原盘删除；同盘移动时，文件的存储位置不变，FILE 记录号也不变，FILE 记录中只有“最后访问时间”和上层文件号发生变化，另外从原来文件夹中删除了该文件的目录项，在新文件夹中增加了该文件号，新旧文件夹的变化都可能引起 B+树的变化。

文件夹移动，文件号不变，文件夹内的各目录项也不变，只是引起文件夹的宿主文件夹 B+树的变化。文件夹的时间属性被刷新了，但是文件夹下的文件夹的时间属性不变。

需要特别注意文件复制后的“最后访问时间”，复制单个文件时，原来位置文件的“最后访问时间”被刷新，而复制文件夹时，原来文件夹下文件的“最后访问时间”变化与否，与文件长度有关，这个长度分界点与操作系统提供的文件缓冲区大小有关。

在 NTFS 文件系统中，文件的增加和减少，可能影响到所在文件夹 B+树的变化，系统将修改文件夹的修改时间和最后访问时间。而在 FAT 系统中，在文件夹创建后，无论文件夹内文件(夹)如何增加和减少，文件夹的修改时间和最后访问时间是不变的。

2.4 文件内容修改产生的痕迹

NTFS 下的文件修改痕迹与 FAT 系统下相似，只是由于 FILE 记录的管理方式，FILE 号被回收后可能很快被重新分配，留下的 FILE 非常少。修改文件内容产生的痕迹与文件的形成方式和修改所用工具软件有关。有的应用软件在文件修改后生成临时文件，一般地，多数应用系统在文件修改后会重新申请存储空间，原来位置的文件内容就被保存了下来，但是不能由操作系统的文件功能访问，而使用专用取证工具可以根据文件特征获取部分或全部文件内容。

2.5 格式化后留下的痕迹

格式化不会清除磁盘中的全部数据簇，无论是否“快速格式化”。格式化将初始化卷的系统参数，如每簇扇区数、MFT、根目录等。没有被覆盖的文件内容，可以根据文件的数据特征部分恢复数据，连续存放的可以完全恢复，不连续存放的文件能否完整地被恢复，取决于文件的存储索引表是否还存在，也就是文件的 FILE 记录是否被覆盖。操作系统给 MFT 预留大约 12.5%的存储空间，在使用过程中根据文件的存储情况增加或减少 MFT 的存储空间，也就是说，MFT 的存储空间可以不连续，其存储分布保存在 MFT 0，实际上 MFT 0 在格式化时并不占有全部预留的空间，而是根据实际存储的文件数来动态调整，占用的部分先被清空，对于没有占用的存储空间，不作任何处理。因此，卷格式化后，先前保存的大量 FILE 记录没有被清除，可以根据留存的 FILE 记录完整地恢复格式化前的文件。这是 FAT 系统所无法做到的，因为 FAT 系统格式化时清除全部 FAT 表，使文件存储的链接关系被破坏。

3 结束语

NTFS 文件系统和 FAT 文件系统在文件操作后留下的痕迹有相似的部分，也有各自不同的部分，还有许多其他痕迹特征需要进一步研究。在计算机取证时需要全面把握，提高证据的利用价值。

参考文献

1 黄步根. 数据恢复与计算机取证[J]. 计算机安全, 2006, (6): 79-80.

2 尤晋元. Windows 操作系统原理[M]. 北京：机械工业出版社, 2001.

3 梁金千, 张 跃. NTFS 文件系统的主要数据结构[J]. 计算机工程与应用, 2003, 39(8): 116-118.

4 黄步根. FAT 系统文件操作痕迹特征分析[J], 计算机工程与应用, 2007, 43(7): 233-235.

（上接第 245 页）

表 1 无生成树模块时数据流量

执行项目	CPE 收包数	CPE 发包数	CPE 转发的广播包	总数
ping 192.168.56.1	9	25	3 896	3 930
	23	41	4 143	4 207
	10	27	4 020	4 057
浏览网页	550	510	7 643	8 703
	60	104	5 332	5 496
	858	780	7 861	9 499

当各个 CM 的 STP 模块启动后，在 CPE 端能正确接收到自主设计的 CM 周期性发出的配置 BPDU。CM2 的 port2 迅速进入阻塞状态，CPE 接收到的广播包明显减少，回传的数据通过率为 0，保证了网络的传输效率和用户的上网速度，CPE 能正常浏览网页。测试结果如表 2 所示(单位：packets，持续时间：1min)。

表 2 有生成树模块时数据流量

执行项目	CPE 收包数	CPE 发包数	CPE 转发的广播包	总数
ping 192.168.56.1	41	49	668	758
	47	54	536	637
	46	53	579	678
浏览网页	577	485	434	1 496
	413	354	678	1 445
	676	569	434	1 679

4 结束语

本文研究并实现了 HFC 网络中的透明网桥和生成树协议。符合 DOCSIS 规范商业用途的 CMTS 和 CM 必须支持网桥和生成树协议。利用生成树协议去除电缆网络和以太网连接中可能出现的网络环路。通过实际系统对生成树模块和透明网桥模块的测试，证明本方法能有效防止网络冗余，形成最优的树状网络拓扑结构，加强了电缆网络传输的有效性和可靠性。

参考文献

1 Cable Television Laboratories, Inc. Data-Over-Cable Service Interface Specifications Radio Frequency Interface Specification[Z]. (2005-09). SP-RFIv1.1-C01-050907. http://www.cablemodem.com.

2 Sdralia V, Tzerefos P, Smythe C. Recovery Analysis of the DOCSIS Protocol After Service Disruption[J]. IEEE Transactions on Broadcasting, 2001, 47(4): 377-385.

3 ANSI/IEEE Std 802.1D-1998 Information Technology Telecommunications and Information Exchange Between Systems Local Area Networks-Part3: Media Access Control (MAC) Bridges[S]. 1998.

4 Lee Wei-Tsong, Chung Kun-Chen, Chu Kuo-Chih, et al. DOCSIS Performance Analysis Under High Traffic Conditions in the HFC Networks[J]. IEEE Transactions on Broadcasting, 2006, 52(1): 21-30.

5 Limb J O, Sala D. A Protocol for Efficient Transfer of Data Over Hybrid Fiber/Coax Systems[J]. IEEE/ACM Transactions on Networking, 1997, 5(6): 872-881.

6 Cable Television Laboratories, Inc. Data-over-cable Service Interface Specifications Cable Modem to Customer Premise Equip- ment Interface[Z]. (2005-09). CM-SP-CMCI-I10-050408. http://www.cablemodem.com.

7 Huang Changcheng, Stodola K. Bridging Core and Edge Networks for Residential Subscribers[J]. IEEE Communications Magazine, 2000, 38(12): 115-121.