Journal of Computer Applications
计算机应用,2014,34(2):417 -420
ISSN 1001-9081
CODEN JYIIDU
2014-02-10
http://www.joca.cn

文章编号:1001-9081(2014)02-0417-04
doi:10.11772/j.issn.1001-9081.2014.02.0417

基于 NTFS 大目录的文件创建方法

吴伟民,林水宾*,江达强,黎海明,苏　庆
(广东工业大学 计算机学院,广州 510006)
(*通信作者电子邮箱 shuibin028@qq.com)

摘　要:在已有文献中,由于不依赖 Windows 应用程序编程接口(API)调用的在新技术文件系统(NTFS)下的文件创建都是在小目录下实现的,因此将对在大目录下创建文件的方法进行研究。运用 B+树遍历,找到索引缓冲区,通过判断所找的索引缓冲区是否存在索引节点,分别将创建好的索引项插入到指定的索引缓冲区位置,将插入索引项的索引缓冲区写入磁盘,完成在大目录下对文件的创建。通过实验,实现了在大目录下的文件创建,证明了方法在不依赖于 Windows API 而实现文件创建的正确性。

关键词:新技术文件系统;大目录;B+树;索引缓冲区;索引节点
中图分类号:TP309.2;TP312　　**文献标志码**:A

Method of creating file based on big directory of NTFS

WU Weimin, LIN Shuibin*, JIANG Daqiang, LI Haiming, SU Qing
(*School of Computer Science, Guangdong University of Technology, Guangzhou Guangdong* 510006, *China*)

Abstract: In the available literatures, creating new files with New Technology File System (NTFS) that does not depend on calling Windows Application Program Interface (API) takes place in small directory. Therefore, a new technological realization of creating files in big directories was proposed in this paper. Firstly, it located the index buffer by traversaling the B+tree. Secondly, by judging whether the index buffer had an index node, it would put the created index entry into the specified location of index buffer respectively. Next, the index buffer inserted by the index was written to disk. Finally, it created new files in the big directories successfully. The experiments prove that the files can be created correctly in a large directory using the new creating technology.

Key words: New Technology File System (NTFS); big directory; B+tree; index buffer; index node

0　引言

新技术文件系统(New Technology File System, NTFS)是微软公司开发的具有较好安全性和容错性的文件系统,其目录结构的研究[1-2]在计算机取证[3]、数据恢复[4]和数据安全等领域具有广泛的应用。基于商业方面考虑,微软并未公开 NTFS 技术细节。

NTFS 运用 B+树文件管理方法来定位文件在磁盘上的位置[5],而由于 B+树索引结构的定义及变化规律较为复杂,使得 NTFS 目录结构的分析相对困难。Carrier[6]、Krten[7]、Russon[8]和戴士剑等[9]对 NTFS 文件系统目录结构进行了详细的分析,并根据 B+树定义,对 NTFS 目录的文件索引结构以及目录中索引项的添加和删除操作进行了理论分析,但并未系统地总结出 NTFS 目录和磁盘上的具体实现机制和在相关文件操作下目录结构的变化规律。刘凯[10-11]和卢琦等[12]分析了 NTFS 大目录产生的原因以及各种存储形式的目录结构,通过实验动态跟踪在大目录下创建和删除文件对大目录结构的影响,动态分析了各种文件操作下的目录在磁盘上的结构的变化规律,但只是局限于应用手工实验的方式在简单情况下实现对文件的创建过程,而未能解决在大多数典型实际情况下创建文件的问题。

因此,本文在前人分析 NTFS 文件目录在磁盘上的存储结构的基础[13-14]上,研究不依赖于 Windows 应用程序编程接口(Application Program Interface, API)调用来实现在大目录下的文件创建方法,完善 NTFS 文件系统对目录和文件的操作和管理,使得非 Windows 用户亦可以在不依赖于 Windows API 甚至在无操作系统环境的前提下操作 NTFS 文件系统。

1　NTFS 文件系统结构分析

在 NTFS 文件系统中,以簇为基本单位对磁盘空间和文件存储进行管理,而簇又分为逻辑簇号(Logical Cluster Number, LCN)和虚拟簇号(Virtual Cluster Number, VCN)。假设定义逻辑簇号为 LCN,每簇扇区数为 SPC,卷的隐含扇区数为 HS,簇的绝对扇区号为 AS,则逻辑簇号与磁盘的扇区号的对应关系为:LCN * SPC + HS = AS。

1.1　主文件表的结构

主文件表(Master File Table, MFT)以文件记录来实现对

收稿日期:2013-07-17;**修回日期**:2013-09-17。　**基金项目**:广州市科技计划项目(2012Y2-00046,2013Y2-00043)。
作者简介:吴伟民(1956 -),男,广东深圳人,教授,CCF 会员,主要研究方向:信息安全、数据结构、可视计算、虚拟机;　林水宾(1989 -),男,福建漳州人,硕士研究生,主要研究方向:文件系统结构、信息安全;　江达强(1991 -),男,广东广州人,主要研究方向:信息安全;　黎海明(1992 -),男,广东珠海人,主要研究方向:文件系统结构;　苏庆(1979 -),男,广东茂名人,讲师,博士研究生,主要研究方向:可视计算、虚拟机、计算机软硬件体系结构。

文件的管理,每个文件记录都对应着不同的文件,大小固定为 1 KB。文件记录由两部分构成,一部分是文件记录头,另一部分是属性列表。

表 1 是 MFT 文件的文件记录头,0x32 偏移后面的是文件记录的属性列表。

表 1 MFT 结构

偏移	尺寸大小/B	描述
0x00	4	MFT 标志,值为:“FILE”
0x04	2	更新序列号
0x06	2	更新序列号个数 +1
0x08	8	日志文件序列号
0x10	2	更新序列号
0x12	2	硬连接数
0x14	2	第一个属性的偏移地址
0x16	2	标志
0x18	4	文件记录的实际长度
0x1C	4	文件记录的分配长度
0x20	8	基本文件记录中的文件索引号
0x28	2	下一个属性 ID
0x2A	2	保留
0x2C	4	该 MFT 记录号
0x30	2	更新序列号
0x32	2	更新数组

1.2 索引缓冲区结构

索引缓冲区是 NTFS 的 B + 树目录管理中重要的结构,每个索引缓冲区在 NTFS 中一般是 4 KB 的大小,其位置和大小由目录的文件记录中 A0H 属性的数据流(Run List)[7-8]定义。

表 2 为索引缓冲区的索引头结构,在索引头后面是索引项。

表 2 索引缓冲区结构

偏移	尺寸大小/B	描述
0x00	4	记录标签,值为:“INDEX”
0x04	2	更新序列号的偏移
0x06	2	更新序列号与更新数组以字为单位的大小(S)
0x08	8	日志文件序列号
0x10	8	当前索引块在目录文件中的虚拟簇号
0x18	4	第一个索引项的偏移
0x1c	4	索引项的总大小
0x20	4	索引项分配大小
0x24	1	如果不是叶子节点,置 1,表示还有子节点
0x25	3	保留
0x28	2	更新序列号
0x2a	$2S-2$	更新序列数组

2 大目录下创建文件的方法实验

首先,通过 B + 树遍历,找到指定大目录的 MFT 文件记录号。通过 B + 树遍历大目录索引项时,有三种情况:

1)找到索引项。通过索引项直接找到 MFT 文件记录号。

2)所找的索引项位于当前索引项的后面。继续往后找,直到找到索引项,再由索引项找到 MFT 文件记录号。

3)所找的索引项在当前索引项的子节点中。由子节点进入索引缓冲区找到索引项,最后找到 MFT 文件记录号。

其次,由大目录的 MFT 文件记录号找到索引缓冲区。记录下 MFT 记录号 90H 属性的 VCN 和 MFT 记录号 A0H 属性的 Run list,再由 VCN 和 Run list 定位到索引缓冲区。

最后,因为索引缓冲区包括多个索引项,每个索引项指向一个目录或者文件,所以通过比较每个索引项的文件名,找到第一个文件名比目标文件名大的索引项。此索引项的开始位置即是创建目标文件索引项的插入点位置。

根据上述中所找到的索引缓冲区是否存在索引节点,即索引缓冲区的 24H 偏移处的值是否为 0:如果为 0,说明没有索引节点;如果为 1,说明有索引节点。所以可以将创建文件的过程分为无索引节点情形下创建文件和有索引节点情形下创建文件两种情形。图 1 是大目录下创建文件的一般流程。

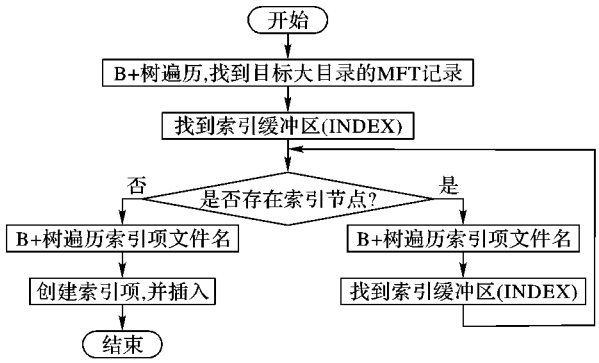

图 1 大目录创建文件的一般流程

2.1 无索引节点情形下创建文件过程

当大目录的索引缓冲区的 24H 偏移处为 0 时,说明索引缓冲区没有索引节点。无索引节点情形下创建文件的过程如下:

首先,通过 B + 树遍历,找到指定大目录的 MFT 文件记录号。以 C:\WINDOWS\System 目录为例,通过 B + 树遍历,找到 System 目录的 MFT 文件记录号,如图 2 所示。

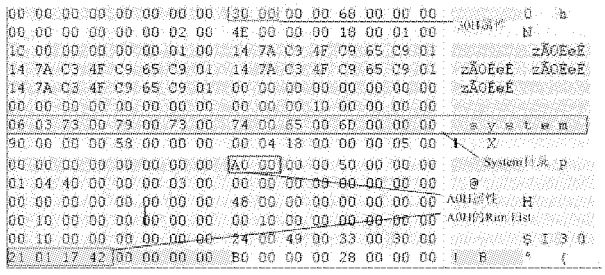

图 2 System 目录的 MFT 文件记录

其次,由大目录的 MFT 文件记录号找到索引缓冲区。在 System 目录的 MFT 中,A0H 属性的 Run List 为 21 01 17 42,通过虚拟簇号与扇区号的转换关系,得到 System 目录的索引缓冲区,如图 3 所示。

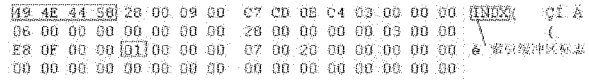

图 3 System 目录的索引缓冲区

最后,通过比较每个索引项的文件名,找到第一个文件名比目标文件名大的索引项。

2.2 有索引节点情形下创建文件过程

当大目录的索引缓冲区的 24H 偏移处的值为 1 时,说明

此索引缓冲区还存在索引节点。有索引节点情形下文件的创建过程为:前面的步骤与无索引节点情形下创建文件的过程是一样的。以 C:\WINDOWS\Media 目录为例,找到 Media 目录的 MFT 文件记录号后,记录 90H 属性的 VCN 以及 A0H 属性的 Run List,如图 4 所示。

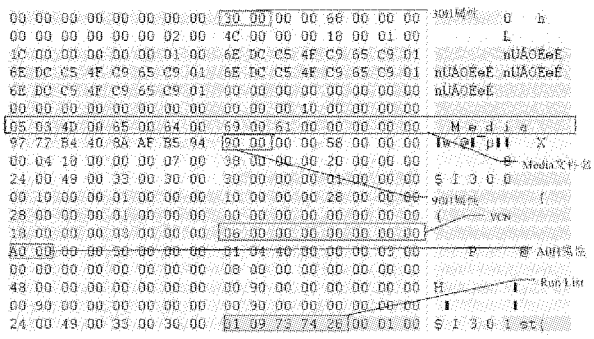

图 4　Media 目录的 MFT 文件记录

通过 A0H 属性的 Run List 得到的虚拟簇号再加上 90H 属性的 VCN 06,得到的簇号再转换为扇区号,即为第一个索引缓冲区的扇区号。如图 5 所示。

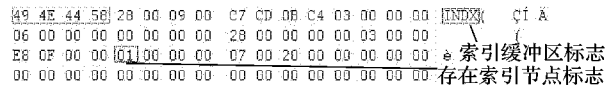

图 5 索引缓冲区

而在找到大目录的索引缓冲区之后,通过 B + 树遍历索引项文件名,找到第一个文件名比目标文件名大的索引项,记录该索引项的 VCN,而 VCN 是该索引项的最后 8 个字节。由 VCN 可以找到下一层索引缓冲区。继续循环上述的过程,直到所找的索引缓冲区无索引节点。后面的文件创建过程和无索引节点情形下创建文件的过程是一样的。

当索引缓冲区中不存在索引节点时,通过比较索引项的文件名,可以直接找到插入索引项的插入点。而当索引缓冲区中存在索引节点时,还需再找下一层索引缓冲区,直到所找的索引缓冲区无索引节点,然后再找到索引项的插入点。

3　实验结果分析

本实验通过对 NTFS 文件系统的目录与文件的结构及管理原理的了解,使用汇编代码实现,并借助硬件仿真器,摆脱了对 Windows API 的依赖而实现了在大目录下创建文件的过程。汇编代码程序的运行时间点位于计算机启动完成后、操作系统引导前。

3.1　实验的环境

系统:Windows 7 Ultimate with Service Pack 1;

文件系统:NTFS;

磁盘分析工具:WinHex 16.7;

实验文件:Adobe Reader. exe。

其中,Adobe Reader. exe 是用于打开和使用在 Adobe Acrobat 中创建 Adobe PDF 的工具。

3.2　实验过程及结果

3.2.1　索引缓冲区没有索引节点的情形

本实验的目的是为了在 C:\WINDOWS\security 大目录下创建 Adobe Reader. exe 文件。其中,security 目录是一个索引缓冲区没有索引节点的大目录。先通过引导扇区 DBR 得到$MFT 的起始扇区号,找到$ROOT 扇区号,再找到 security 目录的 MFT 文件记录,如图 6 所示。

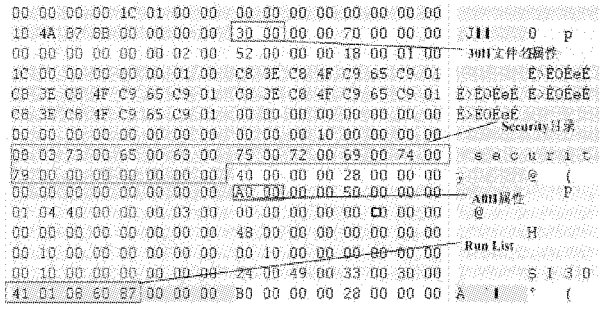

图 6　security 目录的 MFT 记录号

在图 6 中,30H 属性存储的是文件或目录名,其文件或目录名为 security。A0H 属性是索引分配属性,其 48H 偏移处为 Run List 数据流,其数值为 41 01 08 60 87 00。其中,41 是压缩字节,高 4 位为数据流的起始逻辑簇号所占的字节数,低 4 位表示数据流所占用的簇数的字节数。01 表示的是数据流只占用一个簇的大小,08 60 87 00 表示的是起始簇号的相对簇数。由 Run List 的数值可以得到 security 所对应的索引缓冲区,如图 7 所示。图 7 为还没有插入目标文件索引项的索引缓冲区,可以看出在 security 目录下,索引缓冲区的第一个索引项是 Database。

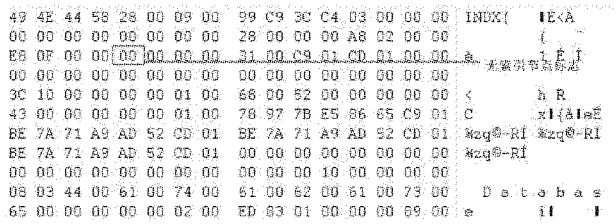

图 7　插入索引项之前 security 目录的索引缓冲区

当插入索引项后,如图 8 所示。

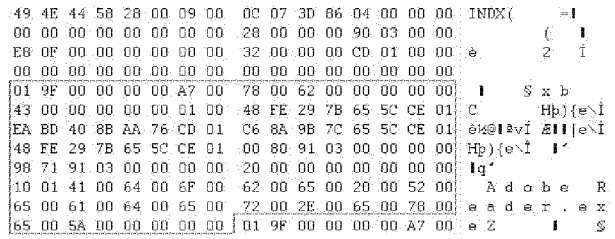

图 8　插入索引项之后 security 目录的索引缓冲区

图 8 的矩形框中的数据为创建文件所插入的索引项,而且该索引项变成了索引缓冲区的第一个索引项,即所插入的位置位于 Database 目录或文件的索引项的前面。可以看出,索引项的插入是按文件或目录名字的字母的大小顺序排序的。其中索引项的前 6 个字节(01 9F 00 00 00 00)就是 Adobe Reader. exe 文件所在的 MFT 文件记录号。得到 Adobe Reader. exe 的 MFT,如图 9 所示。

在图 9 中,30H 属性存储的文件名是 Adobe Reader. exe,说明在没有索引节点的大目录下创建文件成功。

3.2.2　索引缓冲区存在所索引节点的情形

在 C:\WINDOWS\temp 目录下创建 Adobe Reader. exe 文

件,其中,temp 目录的索引缓冲区存在索引节点。查找起始 $MFT的扇区号、$root 扇区号以及大目录的 MFT 号的过程和上面是一样的。当得到 temp 的 MFT 号后,可以找到其索引缓冲区,如 10 所示。

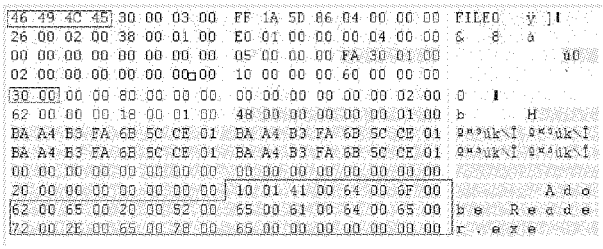

图 9 Adobe Reader. exe 的 MFT 记录号

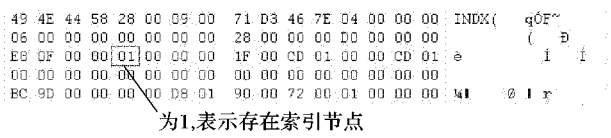

图 10 temp 目录的第一层索引缓冲区

通过 B + 树遍历文件名,找到第一个比目标文件名大的文件名的索引项,通过该索引项得到文件名的 MFT 号,再通过 A0H 属性的 Run List 找到索引缓存区,如图 11 所示。

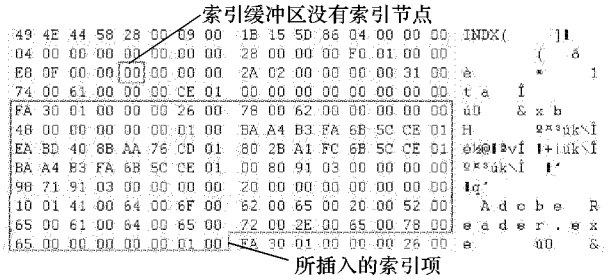

图 11 temp 目录的第二层索引缓冲区及插入的索引项

在图 11 中,索引缓冲区的 24H 偏移处的值为 0,说明没有索引节点。大的矩形框部分是 Adobe Reader. exe 的索引项,表明插入索引项成功。通过索引项,找到 Adobe Reader. exe 的 MFT 记录号,如图 12 所示。

在 MFT 号的 30H 属性中存储的文件名是 Adobe Reader. exe,说明在指定目录下创建了目标文件。所以,在索引缓冲区存在索引节点的大目录下创建文件成功。

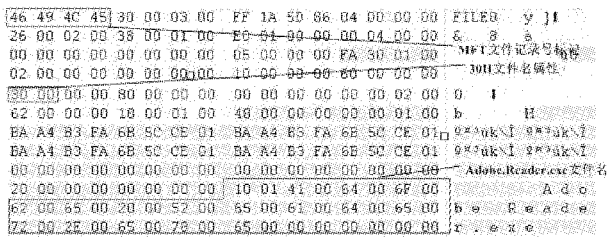

图 12 Adobe Reader. exe 的 MFT 记录号

通过不断的代码调试与测试,实现了大目录下文件的创建。本实验的截图是在代码调试过程中存储和保留的,该实验的结果已通过实验上机验证。

4 结语

本文论述并验证了 NTFS 文件系统下在大目录中创建文件的方法。通过在大目录的索引缓冲区中的指定位置插入索引项,并将该索引缓冲区写回磁盘,实现了用汇编代码完成在大目录下对文件的创建。在下一步工作中,将论述和研究在索引缓冲区中插入索引项后使得索引项的总大小超出了索引缓冲区大小的情况。

参考文献:

[1] WANG L, JU J. The analysis of NTFS file system structure[J]. Computer Engineering and Design, 2007, 28(22): 5457 - 5460. (王兰英, 居锦武. NTFS 文件系统结构分析[J]. 计算机工程与设计, 2007, 28(22): 5457 - 5460.)

[2] LIANG J Q, ZHANG Y. The main data structure of NTFS file system[J]. Computer Engineering and Applications, 2003, 39(8): 113 - 119. (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): 113 - 119.)

[3] HUANG B. The trace analysis of file operations on NTFS system storage medium[J]. Computer Engineering, 2007, 33(23): 281 - 283. (黄步根. NTFS 系统存储介质上文件操作痕迹分析[J]. 计算机工程, 2007, 33(23): 281 - 283.)

[4] MA L. Data reproduced: File system principle solution with best practice of data recovery. [M]. Beijing: Tsinghua University Press, 2009. (马林. 数据重现: 文件系统原理精解与数据恢复最佳实践[M]. 北京: 清华大学出版社, 2009.)

[5] LIU W. Deep reveal data recovery[M]. Beijing: Electronic Industry Press, 2010. (刘伟. 数据恢复深度揭秘[M]. 北京: 电子工业出版社, 2010.)

[6] CARRIER B. File system forensic analysis[M]. Upper Saddle River: Addison-Wesley Professional, 2009.

[7] KRTEN R. The QNX Cookbook: Recipes for programmers[M]. Ottawa: Parse Software Devices, 2008.

[8] RUSSON R. NTFS Documentation[EB/OL]. [2013-03-05]. http://inform.pucp.edu.pe/~inf232/Ntfs/ntfs_doc_v0.5/index.html.

[9] DAI S, TU Y. Data recovery technology[M]. Beijing: Electronic Industry Press, 2005. (戴士剑, 涂彦晖. 数据恢复技术[M]. 北京: 电子工业出版社, 2005.)

[10] WU W, LIU K, JIANG D, *et al*. The dynamic analysis of NTFS B + tree directory structure[J]. Computer Engineering and Design, 2013, 34(4): 84 - 87. (吴伟民, 刘凯, 将达强, 等. NTFS B + 树大目录结构动态解析[J]. 计算机工程与设计, 2013, 34(4): 84 - 87.)

[11] LIU K. The research and design based on NTFS registry forensics tools[D]. Guangzhou: Guangdong University of Technology, 2013. (刘凯. 基于 NTFS 注册表取证工具研究与设计[D]. 广州: 广东工业大学, 2013.)

[12] WU W, LU Q, WANG Z, *et al*. The dynamic analysis of indexing B + tree structure in NTFS directory[J]. Computer Engineering and Design, 2010, 31(22): 185 - 188. (吴伟民, 卢琦, 王振华, 等. NTFS 目录下索引 B + 树结构动态解析[J]. 计算机工程与设计, 2010, 31(22): 185 - 188.)

[13] NTFS Research Group. Disk scan for deleted entries[EB/OL]. [2013-06-20]. http://www.ntfs.corn/disk-scan.htm, 2009.

[14] LIU J, LEE Z J, CHUNG Y C. Dynamic probabilistic packet marking for efficient IP traceback[J]. Computer Networks, 2007, 50(3): 866 - 882.