



北京北亚数据恢复中心
http://www.raid-recovery.org



TEL: 4006-505-808

网站首页

实验室简介

客户服务

技术培训

服务报价

联系我们

您现在的位置: 数据恢复_北京数据恢复|北亚数据恢复中心 4006-505-808 >> 数据恢复文章 >> 数据恢复文
栏 >> 文章正文

报修电话: 4006-505-808

--站内文章检索----

数据恢复

OK

提示: 可按上面表单检索
本站
所有文章及下载资源,
也可通过下面菜单进入文
章、
下载及其他类别!

最新动态

文章中心

下载中心

网络日志

咨询留言

交流论坛

北亚数据恢复中心服务
承诺:

FAT文件系统原理(三)

FAT文件系统原理(三)

更新时间:2004-4-20【字体: 小 大】

4.3 FAT表和数据的存储原则。

FAT表(File Allocation Table 文件分配表), 是Microsoft在FAT文件系统中用于磁盘数据(文件)索引和定位引进的一种链式结构。假如把磁盘比作一本书, FAT表可以认为相当于书中的目录, 而文件就是各个章节的内容。但FAT表的表示方法却与目录有很大的不同。

在FAT文件系统中, 文件的存储依照FAT表制定的簇链式数据结构来进行。同时, FAT文件系统将组织数据时使用的目录也抽象为文件, 以简化对数据的管理。

单击此处查看PDF版全文

★存储过程假想:

我们模拟对一个分区存储数据的过程来说明FAT文件中数据的存储原则。

假定现在有一个空的完全没有存放数据的磁盘, 大小为100KB, 我们将其想象为线形的空间地址。为了存储管理上的便利, 我们人为的将这100KB的空间均分成100份, 每份1KB。我们来依次存储这样几个文件: A. TXT(大小10KB), B. TXT(大小53.6KB), C. TXT(大小20.5KB)。

最起码能够想到, 我们可以顺序的在这100KB空间中存放这3个文件。同时不要忘了, 我们还要记下他们的大小和开始的位置, 这样下次要用时才能找的到, 这就像是目录。为了便于查找, 我们假定用第1K的空间来存储他们的特征(属性)。还有, 我们设计的存储单位是1KB, 所以, A. TXT我们需要10个存储单位(为了说明方便, 我们把存储单位叫做“簇”吧。也能少打点字, 呵呵。), B. TXT需要54个簇, C. TXT需要21个簇。可能有人会说B. TXT和C. TXT不是各自浪费了不到1簇的空间吗? 干嘛不让他们紧挨着, 不是省地方吗? 我的回答是, 如果按照这样的方式存储, 目录中原本只需要记下簇号, 现在还需要记下簇内的偏移, 这样会增加目录的存储量, 而且存取没有了规则, 读取也不太方便, 是得不偿失的。

根据上面所说的思想, 我们设计了这样的图4.3.1所示的存储方式。

- 免费检测
- 免费提供3天备份
- 专业数据恢复工程师提供服务
- 数据恢复前报价，客户确认后工程师开始数据修复
- 数据恢复不成功不收费
- 与客户签订保密协议，对客户的数据严格保密
- 整个恢复过程不会对客户的原盘有任何的写操作，以确保原盘的数据完全。

图4.3.1 整个100KB空间

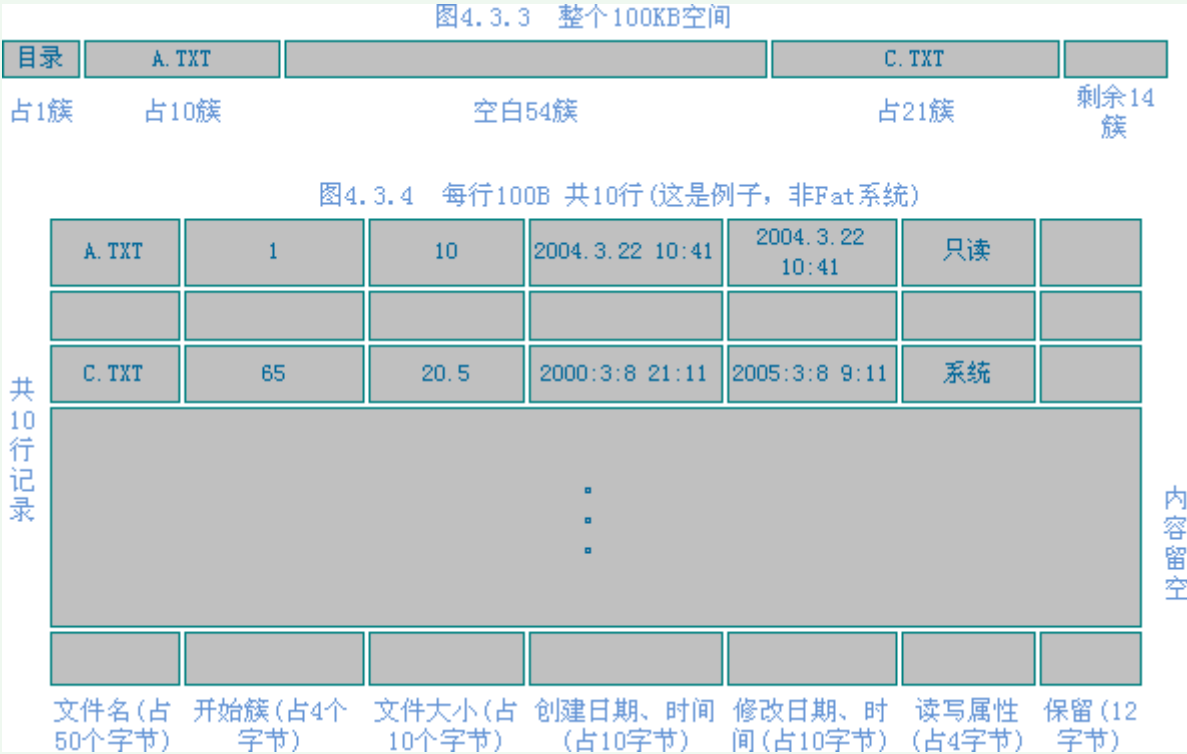
目录	A.TXT	B.TXT	C.TXT	
占1簇	占10簇	占54簇	占21簇	剩余14簇

我们再考虑如何来写这三个文件的目录。对于每个文件而言，一定要记录的有：文件名，开始簇，大小，创建日期、时间，修改日期、时间，文件的读写属性等。这里大小能不能用结束簇来计算呢？一定不能，因为文件的大小不一定是整数个簇的大小，否则的话像B.TXT的内容就是54KB的内容了，少了固然不行，可多了也是不行的。那么我们怎么记录呢？可以想象一下。为了管理上的方便，我们用数据库的管理方式来管理我们的目录。于是我把1KB再分成10份，假定开始簇号为0，定义每份100B的各个位置的代表含义如图4.3.2

图4.3.2 每行100B 共10行(这是例子，非Fat系统)

共 10 行 记 录	A.TXT	1	10	2004.3.22 10:41	2004.3.22 10:41	只读	有效记录
	B.TXT	11	53.6	1949:10:1 12:0	2003.8.22 20:40	隐藏	
	C.TXT	65	20.5	2000:3:8 21:11	2005:3:8 9:11	系统	
	· · ·						内容留空
	文件名(占50个字节)	开始簇(占4个字节)	文件大小(占10个字节)	创建日期、时间(占10字节)	修改日期、时间(占10字节)	读写属性(占4字节)	保留(12字节)

这样设计的结构绝对可以对文件进行正确的读写了。接着让我们设计的文件系统工作吧。先改动个文件，比如A.TXT，增加点内容吧！咦？增加后往哪里放呀，虽然存储块的后面有很多空间，但紧随其后B.TXT的数据还顶着呢？要是把A.TXT移到后边太浪费处理资源，而且也不一定解决问题。这个问题看来暂时解决不了。那我们换个操作，把B.txt删了，b.txt的空间随之释放。这时候空间如图4.3.3，目录如图4.3.4



这个操作看来还可以, 我们接着做, 在存入一个文件D.txt(大小为60.3KB), 总共100簇的空间只用了31簇, 还有68簇剩余, 按说能放下。可是? 往那里放呢? 没有61个连续的空间了, 目录行没办法写了, 看来无连续块存储暂时也不行。

你一定能够想到我们可以在连续空间不够或增加文件长度的时候转移影响我们操作的其他文件, 从而腾出空间来, 但我要问你, 那不是成天啥也不要干了, 就是倒腾东西了吗?

看来我们设计的文件系统有致命的漏洞, 怎么解决呢?。。。。。

其实可以这样解决:
首先我们允许文件的不连续存储。目录中依然只记录开始簇和文件的大小。那么我们怎么记录文件占用那些簇呢, 以文件映射簇不太方便, 因为文件名是不固定的。我们换个思想, 可以用簇来映射文件, 在整个存储空间的前部留下几簇来记录数据区中数据与簇号的关系。对于上例因为总空间也不大, 所以用前部的1Kb的空间来记录这种对应, 假设3个文件都存储, 空间分配如图4.3.5, 同时修改一下目录, 如图4.3.6

文件分配表	目录	A. TXT	B. TXT	C. TXT	
第0簇	第1簇	第2~11簇	第12~65簇	第66~86簇	第87~99簇

A.TXT	2	10	2004.3.22 10:41	2004.3.22 10:41	只读	
B.TXT	12	53.6	1949:10:1 12:0	2003.8.22 20:40	隐藏	
C.TXT	66	20.5	2000:3:8 21:11	2005:3:8 9:11	系统	
⋮						
文件名(占 50个字节)	开始簇(占4 个字节)	文件大小(占 10个字节)	创建日期、时间 (占10字节)	修改日期、时间 (占10字节)	读写属性(占 4字节)	保留(12 字节)

用图 4.3.7 的组织方式是完全可以实现对文件占有簇的记录。但还不够效率。比如文件名在文件分配表中记录太多, 浪费空间, 而实际上在目录中已经记录了文件的开始簇了。所以可以改良一下, 用链的方式来存放占有簇的关系, 变成图 4.3.8 的组织方式。

图4.3.7 文件分配表

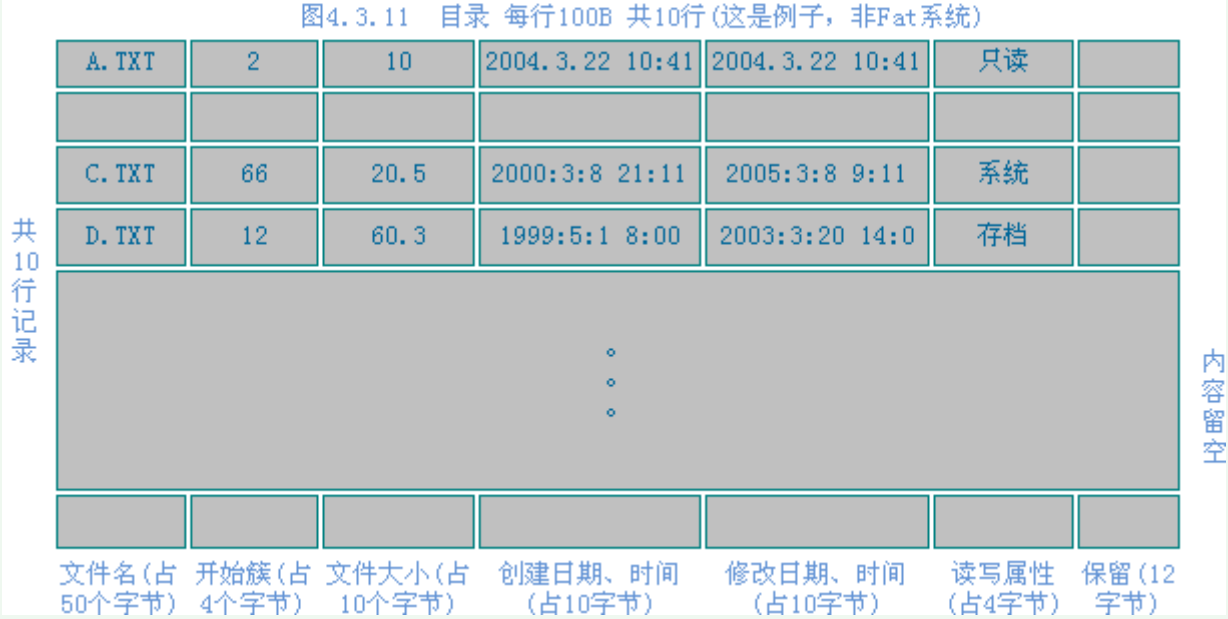
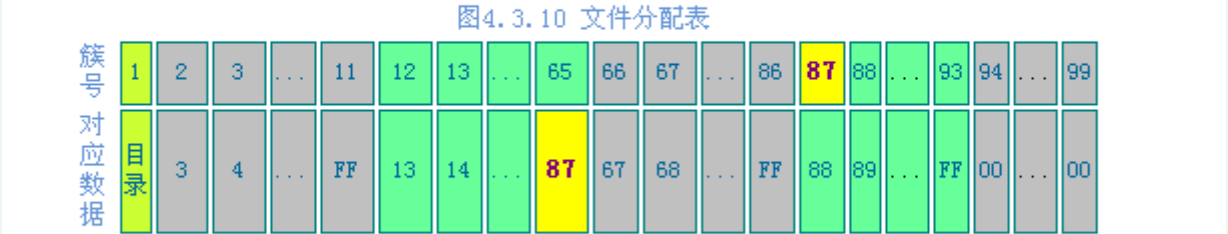
簇号	1	2	3	...	11	12	13	...	65	66	67	...	86	87	...	99
对应数据	目录	A.TXT (1)	A.TXT (2)	...	A.TXT (10)	B.TXT (1)	B.TXT (2)	...	B.TXT (54)	C.TXT (1)	C.TXT (2)	...	C.TXT (21)			

图4.3.8 改良后的文件分配表

簇号	1	2	3	...	11	12	13	...	65	66	67	...	86	87	...	99
对应数据	目录	3	4	...	FF	13	14	...	FF	67	68	...	FF	00	...	00

参照图4.3.8来理解一下文件分配表的意义。如文件a.txt我们根据目录项中指定的a.txt的首簇为2，然后找到文件分配表的第2簇记录，上面登记的是3，我们就能确定下一簇是3。找到文件分配表的第3簇记录，上面登记的是4，我们就能确定下一簇是4.....直到指到第11簇，发现下一个指向是FF，就是结束。文件便丝毫无误读取完毕。

我们再看上面提到的第三种情况，就是将b.txt删除以后，存入一个大小为60.3KB的d.txt。利用簇链可以很容易的实现。实现后的磁盘如图4.3.9 4.3.10 4.3.11



上面是我们对文件存储的一种假设, 也该揭开谜底的时候了。上面的思想其实就是fat文件系统的思想的精髓(但并不是, 尤其像具体的参数的意义与我们所举的例子是完全不同的。请忘掉上边细节, 努力记忆下边)。

[单击此处查看PDF版全文](#)

- 上一篇文章: FAT文件系统原理(二)

● 下一篇文章： FAT文件系统原理(四)

【关闭窗口】

©2001-2006 北京
北亚数据恢复中心

站内搜索

文章标题

所有栏目

关键字

搜索

首页 联系我们 加入收藏 版权申明 文章地图 下载地图 RSS生成 XML生成 友情链接

北亚数据恢复中心

全国统一客服电话：4006-505-808

北京部：北京市海淀区中关村大街11号E世界A座832B

京ICP备09039053号