[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/30 (2006.01)

[12] 发明专利说明书

专利号 ZL 200610144195.2

[45] 授权公告日 2008 年 7 月 23 日

[11] 授权公告号 CN 100405373C

[22] 申请日 2006.11.29

[21] 申请号 200610144195.2

[73] 专利权人 北京中星微电子有限公司

地址 100083 北京市海淀区学院路 35 号世宁大厦 15 层

[72] 发明人 高占东 蒋长洪

[56] 参考文献

US2005010610A1 2005.1.13

JP2006309536A 2006.11.9

CN1766845A 2006.5.3

CN1773509A 2006.5.17

CN1555532A 2004.12.15

审查员 王雪莲

[74] 专利代理机构 北京安信方达知识产权代理有限公司

代理人 龙 洪 霍育栋

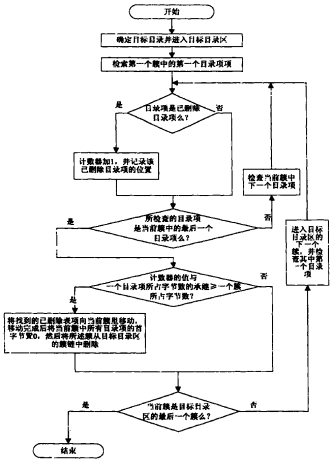
权利要求书 2 页 说明书 11 页 附图 2 页

[54] 发明名称

一种文件系统的目录项整理方法

[57] 摘要

一种文件系统的目录项整理方法，包括：确定目标目录并进入该目标目录区开始检查目录项；逐个判断目标目录区中未检查的目录项为有效目录项还是已删除目录项，当找到的已删除目录项够装满一个簇时，将找到的已删除目录项的位置调换到当前簇里，所述当前簇为目前查到的最后一个簇；调换完成后将该簇中所有目录项的首字节置 0，然后将该簇从目标目录区的簇链中删除；如果目标目录区中目录项未检查完则继续检查，否则结束整理。采用本发明的技术方案后，可以根据需要指定目录进行目录项的整理，通过清理已删除目录项，使有效目录项可以连续存放，从而减少需要比较的目录项数目，提高嵌入式系统中 FAT 文件系统的操作效率，最终提升嵌入式系统的性能。



1、一种文件系统的目录项整理方法，其特征在于，包括:

（a）确定目标目录并进入该目标目录区开始检查目录项;

（b）逐个判断目标目录区中未检查的目录项为有效目录项还是已删除目录项，当找到的已删除目录项够装满一个簇时执行（c）;

（c）将找到的已删除目录项的位置调换到当前簇里，所述当前簇为目前查到的最后一个簇；调换完成后将该簇中所有目录项的首字节置0，然后将该簇从目标目录区的簇链中删除；执行（d）;

（d）如目标目录区中目录项未检查完则返回（b），否则结束整理。

2、如权利要求1所述的方法，其特征在于：步骤（b）中，用计数器记录找到的已删除目录项的个数，当该个数与目录项空间大小的乘积大于或等于一个簇的空间大小时，认为找到的已删除目录项够装满一个簇。

3、如权利要求1所述的方法，其特征在于，步骤（b）进一步细化为:

（b1）判断所检查的目录项是否为已删除目录项，执行（b2）;

（b2）判断所检查的目录项是否为当前簇中的最后一个目录项，如果是则执行（b3）；否则检查当前簇里的下一个目录项并执行（b1）;

（b3）判断找到的已删除目录项是否够装满一个簇，够则执行（c），不够则执行（d）。

4、如权利要求3所述的方法，其特征在于，步骤（c）中，将找到的已删除目录项的位置调换到当前簇里，实现方法是：将当前簇及目标目录区的簇链中在其之前的簇里的有效目录项依次上移，与找到的已删除目录项对调位置；当前簇及其之前的簇里的有效目录项都上移后，移动完成。

5、如权利要求1所述的方法，其特征在于，步骤（c）中，将所述簇从目标目录区的簇链中删除，其实现方法是：在文件分配表中，用所述簇的链表指针取代前一个簇的链表指针，并将当前簇的链表指针置0；所述前一个簇为目标目录区的簇链中在所述簇之前的一个簇。

6、如权利要求3所述的方法，其特征在于，步骤（d）中，判断当前簇

是否为目标目录区的最后一个簇，如果是则目标目录区中目录项已检查完，结束整理；否则目标目录区中目录项未检查完，进入目标目录区的下一个簇并检查其中的第一个目录项，返回（b1）。

7、如权利要求 4 所述的方法，其特征在于，目标目录区中各目录项按其在整个目标目录区中的排列次序有相应的逻辑编号；当所检查的目录项为已删除目录项时，使用数组记录找到的已删除目录项的位置，将数组元素依次赋值为每个找到的已删除目录项的编号；在有效目录项依次上移，与找到的已删除目录项对调位置时，将数组中相应元素的值由已删除目录项的编号更改为所上移的有效目录项上移前的编号。

8、如权利要求 1 所述的方法，其特征在于，确定目标目录的方法包括：由使用者输入目录名或预设在系统中。

一种文件系统的目录项整理方法

技术领域

本发明涉及文件系统，特别涉及文件系统的目录项整理。

背景技术

现在很多嵌入式手持设备如手机，MP3/MP4 播放器等都开始支持外部存储器接口，如 SD 卡标准接口，NandFlash 接口等。这些嵌入式手持设备都使用 FAT (file allocation table 文件分配表)文件系统作为访问外部存储器的接口，根据外部存储器容量不同使用 FAT12/FAT16/FAT32 文件系统进行文件读写等操作。

非易失性存储器上的文件系统一般具有 BIOS（基本输入输出系统）参数块、文件分配表（FAT 表）、根目录和数据区，这样的文件系统一般称为 FAT 文件系统，目前包括 FAT12、FAT16 和 FAT32 三种，文件系统中以簇作为基本的储存单位。其中：

数据区用于存放文件内容的数据。

BIOS 参数块记录了每扇区字节数、每簇扇区数、每个 FAT 的扇区数、扇区总数、根目录项数等等参数。

根目录里记录着每个文件及目录的目录项，也称为目录登记项或表项；目录项中记录着文件或目录名、文件或目录的首簇号，文件大小、文件创建的日期等描述逻辑文件数据结构的数据。根目录和各层子目录都有自己的 FDT（文件目录表）用于存放本目录下各文件及目录的目录项。

FAT 表是记录每个逻辑文件在数据区的物理存储位置的链表，一般以簇为单位；这里的逻辑文件包括文件及目录。FAT 表区按 1 个半字节（FAT12）、2 个字节（FAT16）、4 个字节（FAT32）划分为多个区间（也可称之为单元、元素），每一区间对应于数据区的一个簇，按顺序为这些区间编号，称

为簇号，该簇号除了是一个数据外，还对应于 FAT 表区的某一区间及数据区的一个簇。

一个逻辑文件会占用一个或以上的簇，，文件存放在数据区的簇中的是数据；而目录存放在数据区的簇中的是目录项，这些目录项记录着该目录下所有子目录及文件的信息。当占用一个以上的簇时，逻辑文件是装满一个簇后，再往下一个簇里装。由于簇的存放顺序在物理地址上未必连续，因此在逻辑文件的 FAT 表中，每个簇号对应的区间都存放有该逻辑文件数据所在的下一个簇的簇号，即每个簇号都有一个链表指针指向该逻辑文件数据所在的下一个簇，从而构成一个链表，称为簇链；如果一个簇号的区间中存放的不是下一个簇号而是 FF，则表示这个簇是逻辑文件的最后一个簇。这样，只要知道逻辑文件首簇号，即可找到一个簇链，进而得到该逻辑文件所有数据的存储位置。根目录和各层子目录都有各自独立的簇链。

随着用户进行创建文件、删除文件操作的增多，外部存储器中文件系统的目录项——包括文件和目录的——也会不断的增加，直到达到一个目录下的最大目录项数目．一个目录在数据区中对应的存储空间即该目录的目录区，目录中的目录项即存放在其目录区里．在文件系统的目录区中，随着已删除目录项和新目录项的增加，目录区的结构会越来越庞大，当删除一个文件或目录时，除了将其在数据区的数据删除以外，还会将其目录项更改，一般是将其目录项的首字节改为 E5，这时其目录项就成为一个已删除目录项，但这个已删除目录项实际上还存在于目录区的原来位置．在进行文件或目录的查找或创建操作时，需要将目标文件或目录的名字与目录区中的各目录项进行比较，由于目录项连续存放，比较就是按物理地址依次去比较，因此也会对已删除目录项进行比较，如果已删除目录项很多，这种无用的查找操作就会越多，造成更多的资源浪费．例如：在某个嵌入式系统中的外部存储设备文件系统中，在一个有用目录项数目为 100，已删除目录项数目为 0 的目录，创建一个新文件的时间大约是 150 毫秒；但在一个有用目录项数目为 100，已删除目录项数目为 6000 的目录，创建一个新文件的时间大约为 1400 毫秒，所用时间大约相差 10 倍，在已删除目录项上浪费的时间占用了系统 90%的资源．

嵌入式系统的资源和性能都比较有限，这种情况下，在进行文件操作如创建、删除、查找等操作时会非常耗时，导致产品整体性能下降，无法满足客户需求。

发明内容

针对上述不足，本发明提出了一种文件系统的目录项整理方法，该方法可以清除指定目录区里的已删除目录项，使有效目录项集中存放。

本发明采用的技术方案是:

一种文件系统的目录项整理方法，其特征在于，包括:

（a）确定目标目录并进入该目标目录区开始检查目录项;

（b）逐个判断目标目录区中未检查的目录项为有效目录项还是已删除目录项，当找到的已删除目录项够装满一个簇时执行（c）;

（c）将找到的已删除目录项的位置调换到当前簇里，所述当前簇为目前查到的最后一个簇；调换完成后将该簇中所有目录项的首字节置0，然后将该簇从目标目录区的簇链中删除；执行（d）;

（d）如目标目录区中目录项未检查完则返回（b），否则结束整理。

进一步地，步骤（b）中，用计数器记录找到的已删除目录项的个数，当该个数与目录项空间大小的乘积大于或等于一个簇的空间大小时，认为找到的已删除目录项够装满一个簇。

进一步地，步骤（b）进一步细化为:

（b1）判断所检查的目录项是否为已删除目录项，执行（b2）;

（b2）判断所检查的目录项是否为当前簇中的最后一个目录项，如果是则执行（b3）；否则检查当前簇里的下一个目录项并执行（b1）;

（b3）判断找到的已删除目录项是否够装满一个簇，够则执行（c），不够则执行（d）。

进一步地，步骤（c）中，将找到的已删除目录项的位置调换到当前簇里，实现方法是：将当前簇及目标目录区的簇链中在其之前的簇里的有效目录项依次上移，与找到的已删除目录项对调位置；当前簇及其之前的簇里的

有效目录项都上移后，移动完成。

进一步地，步骤（c）中，将所述簇从目标目录区的簇链中删除，其实现方法是：在文件分配表中，用所述簇的链表指针取代前一个簇的链表指针，并将当前簇的链表指针置0；所述前一个簇为目标目录区的簇链中在所述簇之前的一个簇。

进一步地，步骤（d）中，判断当前簇是否为目标目录区的最后一个簇，如果是则目标目录区中目录项已检查完，结束整理；否则目标目录区中目录项未检查完，进入目标目录区的下一个簇并检查其中的第一个目录项，返回（b1）。

进一步地，目标目录区中各目录项按其在整个目标目录区中的排列次序有相应的逻辑编号；当所检查的目录项为已删除目录项时，使用数组记录找到的已删除目录项的位置，将数组元素依次赋值为每个找到的已删除目录项的编号；在有效目录项依次上移，与找到的已删除目录项对调位置时，将数组中相应元素的值由已删除目录项的编号更改为所上移的有效目录项上移前的编号。

进一步地，确定目标目录的方法包括：由使用者输入目录名或预设在系统中。

采用本发明的技术方案后，可以根据需要指定目录区进行目录项的整理，通过清除该目录区里的已删除目录项，使有效目录项可以连续存放，从而减少需要比较的目录项数目，提高嵌入式系统中 FAT 文件系统的操作效率，最终提升嵌入式系统的性能；同时还可以将已删除目录项所占用的存储空间释放，提高了系统资源的利用率。

附图说明

图 1 是本发明的应用实例中目标目录区的结构示意图。

图 2 是本发明的文件系统目录项整理方法具体实施的流程图；

具体实施方式

下面将结合附图和实施例对本发明的技术方案进行更详细的说明。

为了方便后面的说明，下面再简单介绍一下目录区的结构，如图 1 所示。一个目录区占用一个或以上的簇，这些簇的物理地址不一定连续，簇之间的联系及先后顺序是根据目录区在 FAT 表中的簇链决定的，比如该目录区共占 3 个簇：1、2 和 3；簇链里 1 的下一个簇是 2，即 1 的链表指针指向 2；2 的下一个簇是 3 ，即 2 的链表指针指向 3。

根目录的首簇号记录在 BIOS 参数块中。而子目录的首簇号记录在该子目录的上一层目录的 FDT 中。知道了首簇号就等于知道了目录的入口地址。

每个簇里存放了多个目录项，簇空间的大小是目录项空间大小的整数倍，对于同个系统而言，簇和目录项的大小都是固定值，因此一个系统中，每个簇能装多少个目录项也是固定的。除了最后一个簇以外，目录区中其它簇都是装满目录项的；簇内各目录项的存放在物理地址上是连续的，对于同一个簇中的目录项而言，将前面的目录项的地址直接物理地址累加后就可以得到后面的目录项的地址。如果目录项的空间大小为 32 字节，则簇中的第一个目录项就是该簇所对应的数据区的第一个 32 字节。由于目录区所占空间的大小未必是簇的整数倍，因此最后一个簇中往往留有一些未使用的目录项，其首字节为 0，当遇到这样的目录项时，文件系统程序会认为这是目录区中最后一个目录项。

本发明提供了一种文件系统目录项的整理方法，应用于嵌入式系统，这里的 FAT 文件系统包括 FAT12、FAT16 和 FAT32 文件系统；由于实际应用中目录项的空间大小通常为 32 字节，所以本文中均按一个目录项占 32 字节的情况来说明本发明的技术方案；如果出现目录项的空间大小不为 32 字节的情况，则将表述中的 32 字节更改为目录项的空间大小即可。

所述方法包括以下步骤，如图 2 所示：

a0：确定目标目录——即需要对其中的目录项进行整理的目录；

确定的方式可以根据需要设定，比如由使用者在需要时输入想整理的目录的名称；这里的目录指的是存储器文件系统中的目录，可以是根目录，也可以是各层子目录；再比如，还可以预设在系统的定时任务中，让系统在到达预定时间后，对预定的目录进行整理。

确定后进入目标目录区中的第一个簇，同时设置一个计数器并将其初始数值设为 0。

a1：开始检查目标目录区里第一个簇中的第一个目录项；

b1：当所检查的目录项的首字节为 E5 时，表示该目录项为已删除目录项，将计数器的值加 1，执行 b2。否则该目录项为有效目录项——即已经使用还未删除的目录项，也执行 b2。

在该步中，可以记录每个找到的已删除目录项。为方便后面的说明，在这里假设一个簇能装 N 个目录项。

簇中目录项按其在整个目标目录区中的排列次序有相应的逻辑编号，这里的逻辑编号实际上是一种虚拟的编号，只为了方便记录而设定，比如第一个簇中目录项编号为 0，1，...，N-1，第二个簇中目录项编号为 N，N+2，...，2N-1，以此类推。然后定义一个有 2N-1 个元素的数组变量来记录所发现的已删除目录项的位置，每找到一个已删除目录项，就依次将这个数组的一个元素的赋值为所述已删除目录项的编号，从而记录所发现的已删除目录项的位置。

b2：判断所检查的目录项是否为当前簇中的最后一个目录项，根据 a0 中的计算，一个簇中能装 N 个簇，则每个簇中的第 N 个目录项就是该簇的最后一个目录项；如检查的是当前簇中第 N 个目录项时，执行 b3；如果不是则检查下一个目录项并返回 b1

b3：判断找到的已删除目录项是否能装满一个簇；当计数器的数值大于或等于 N 时，或是当目录项的空间大小——即 32 字节与计数器的值的乘积大于或等于一个簇的空间大小时，表示找到的已删除目录项已经够装满一个簇了，执行 c，在这里或在步骤 c 里将计数器的值减去 N，所述计数器的值即代表找到的已删除目录项的个数；否则执行 d。

c：首先将找到的已删除目录项向当前簇里移动，实现方法是将目标目录区中当前簇——即目前查到的最后一个簇——及目标目录区的簇链中在其之前的各簇中的全部有效目录项依次上移。所述“依次”是指上移是按有效目录项在簇中的排列顺序一个个进行的，因为有的长文件名的目录项不止

一个，而是连在一起的两个或几个，所以有效目录项之间的顺序不能改变。

上移的实际操作是：每次用一个有效目录项取代或者说覆盖其前面第一个已删除目录项——这里寻找“前面第一个已删除目录项”的范围，是从目标目录区第一个簇的第一个目录项到该有效目录项，也就是说有效目录项覆盖的也可能是其它簇里的已删除目录项。如果没有采用数组记录，则需要边移动边判断哪些目录项是已删除的；如果采用了数组记录，则在移动时可以直接使用数组中的信息对目录项进行移动。

覆盖后，如未采用数组记录则将原有效目录项的首字节改为 E5，如采用了数组记录，则将数组中相应元素的值由被覆盖的已删除目录项的编号改为所上移的有效目录项上移前的编号。从效果上可以把每个有效目录项上移的过程看作是有效目录项与已删除目录项对调位置；接着再对下一个有效目录项进行相同操作；如果一个有效目录项前没有已删除目录项时则去对下一个有效目录项操作，直到将当前簇及其之前各簇中的有效目录项都上移完。从效果上看，上移的过程就相当于把已删除目录项向当前簇里移动。

移动完成后，当前簇中存放就都是已删除目录项了，装不下的已删除目录项这时存放在当前簇的前一个簇的最后。当前簇为目标目录区中的最后一簇并且其存放有未使用目录项时，情况会有一点不同，因为未使用目录项不上移，因此移动完成时，该簇中存放的是已删除目录项和未使用目录项。当然，在实际应用中也可以让未使用目录项一起上移。

这时，对当前簇中所有目录项进行清 0 操作，即将它们的首字节置 0；清 0 的结果就是在数据区里彻底清除这些目录项，或者说是释放了这些目录项所占用的存储空间，使其原来的存储空间可再被利用。

然后更新目标目录区在 FAT 表中的链表指针，将当前簇的簇号从目标目录区的簇链中删除，步骤是用当前簇在 FAT 表中对应区间的内容覆盖前一个簇对应区间的内容，并将当前簇的对应区间置 0，或者说是用当前簇的链表指针取代前一个簇的链表指针，并将当前簇的链表指针置 0。前一个簇即目标目录区的簇链中在当前簇之前的一个簇。比如原先目标目录的簇链是：1 的下一个簇为 2，2 的下一个簇为 3，此时 2 已装满已删除目录项，那么改变目标目录区在 FAT 表中的链表指针，将 2 对应区间中原本所存的簇

号 3 放进 1 对应的区间，即将 1 由指向 2 改为指向 3，其实质就是将 2 屏蔽。

如果不进行屏蔽的话，当文件系统程序遇到该簇中的目录项时，会误以为后面没有其它目录项，从而造成错误操作；屏蔽后当前簇重新成为一个可用簇，也就是说文件系统将可以把新的数据存储在该簇里。而且此后当在目标目录中创建或查找文件需要对目录项进行比较时，就不会再查找到该簇，从而也就无须对该簇里的已删除目录项再进行比较了。

执行 d。

d：如果当前簇是目标目录区中的最后一个，那么该簇所对应的区间中存放的不是下个簇的簇号，而是 FF，此时结束整理任务；如果不是 FF 则进入下一个簇开始检查第一个目录项，返回 b1。

结束整理任务后，目标目录中或许还存在有若干已删除目录项，这些目录项由于不能或不够装满一个簇，所以没能被删除；如果有计数器的话，这时计数器的值不为 0。但是，这些剩余的已删除目录项已经对系统的影响非常小，基本可以忽略。

上述为本发明的最优实施方式，在不背离本发明精神及其实质的情况下，熟悉本领域的技术人员可根据本发明作出各种相应的变形，但这些相应的变形都应属于本发明所附权利要求及其等效物所保护的范围内，比如可以每检查到一个已删除目录项就去判断找到的已删除目录项是否够装满一个簇；再比如可以每检查到一个已删除目录项就将其后面所有未检查的目录项依次上移；再比如上移时可以除了当前簇及其之前的簇里的有效目录项，还把当前簇之后的簇里的所有目录项都上移，移完后屏蔽目标目录区里的最后一个簇。

下面用本发明的一个应用实例进一步加以说明。

本实例采用由用户输入目录名的方式来确定目标目录。在本实例中，实现本发明技术方案的软件系统包括以下软件组件：

组件 1，用户接口应用程序，为用户提供应用接口，用户输入目录名来指定文件系统中的一个目标目录区；

组件 2，目录清理组件，接收用户输入的目录名，对目标目录区中的目

录项进行检查，如果发现已删除目录项并积累到目录项数目之和等于一个簇的大小，就会对这部分相关目录项进行整理，删除这些已删除目录项，整理其他有用目录项；

组件 3，FAT 文件系统组件，提供对 FAT 文件系统的基本操作软件接口，如目录访问接口，文件访问接口等；及

组件 4， 存储器控制驱动程序，用于控制存储器，提供对存储器进行复位、读扇区、写扇区等操作的软件接口。

其中，组件 1 是面对用户的应用层面的接口；组件 2 为组件 1 服务，提供了具体程序；而组件 3 为组件 2 提供了对 FAT 文件系统操作的接口；组件 4 是最底层的功能模块，它为组件 3 提供了操作物理层的接口。在进行清理目录项时，上述 4 个组件作为一个整体，在不同层面共同实现了这一操作。

本应用实例中，需要清理的目录区结构如图 2 所示，其中簇 1 为该目录区的首簇，簇 2 是该目录区的第二个簇，以此类推，簇 m 是目录区的最后一个簇；在每个簇里有若干目录项，以簇 1 为例，目录项 1 是簇 1 中的第一个簇，以此类推，目录项 n 是簇 1 中的最后一个簇，即一个簇能存放 n 个目录项；假设目录项 1、目录项 2、目录项 3、目录项 6 为有效目录项，目录项 4 和目录项 5 为已删除目录项，

在上述目录区中清理目录项的步骤如下：

首先，用户输入目录名，确定要清理的目标目录区，这部分工作由组件 1 完成。

组件 2 接收该目录名，并从 BIOS 块中查找到根目录的首簇号，进入根目录；如果目标目录不是根目录则在根目录的 FDT 中查找目标目录的目录项，得到目标目录区的首簇号。在本应用实例中采用数组记录找到的已删除目录项的位置，各目录项的逻辑编号为：簇 1 中目录项 1 的编号为 0，以此类推，目录项 n 的编号为 n-1；簇 2 中第一个目录项的编号为 n，以此类推，簇 m 中最后一个簇的编号为 $m \times n-1$；然后定义一个有 2n-1 个元素的数组用来记录。

进入目标目录区中的首簇，即簇 1，开始检查第一个目录项，即目录项

1，同时设置一个数值为 0 的计数器；每检查一个目录项后即判断其是否为簇 1 中的最后一个目录项，不是则检查下一个目录项。目录项 1 为有效目录项；接着检查目录项 2；以此类推。当检查到目录项 4 时，发现该目录项的首字节为 E5，这意味着目录项 4 为已删除目录项；将计数器的值加 1 并在数组里记录目录项 4 的位置，记录方法是将数组中第一个元素赋值为目录项 4 的编号 3。

然后检查目录项 5；目录项 5 也是已删除目录项，将计数器的值再加 1 并记录目录项 5 的位置，即将数组中第二个元素赋值为目录项 5 的编号 4；接着检查目录项 6；以此类推，直到检查完簇 1 中最后一个目录项，即目录项 n。

这时，判断找到的已删除目录项是否够装满一个簇；如果计数器中的值乘以 32 字节的乘积小于一个簇的空间大小，则说明不够装满一个簇，那么就对簇 1 是不是目标目录区的最后一个簇进行判断，不是则进入簇 2 检查第一个目录项，并重复上述步骤。每检查完一个簇里的目录项，就判断一次已删除目录项是否够装满一个簇，不够就继续检查下面的簇。假设检查完簇 x 后，共发现了 n+y 个已删除目录项，即计数器的值为 n+y，已经够装满一个簇了，这时执行下面的步骤。

将簇 1 到簇 x 中的有效目录项依次上移，根据数组的记录，第一个找到的已删除目录项为目录项 4，则目录项 1、目录项 2 和目录项 3 不用移动；用下一个有效目录项，即目录项 6 的内容覆盖目录项 4，然后将数组中第一个元素的值由目录项 4 的编号 3 改为目录项 6 的编号 5，使原先的目录项 6 成为将来准备被覆盖的“已删除目录项”；以此类推，直到将簇 x 中的有效目录项都上移完。此时，簇 x 中就都是已删除目录项了，并且簇 x-1 里还有 y 个已删除目录项。

此时，对簇 x 进行清 0 操作，即将其中所有目录项的首字节置 0；然后更改目标目录区中 FAT 表中的链表指针，将簇 x 从目标目录区的簇链中删除，这里即在簇链中让簇 x-1 指向簇 x+1，并把簇 x 的链表指针置 0。还要将计数器的值减去 n，即变为 y。

由于 FAT 表中簇 x 的区间中是下一个簇的簇号，而不是 FF，因此按照

前面的步骤继续检查下一个簇里的目录项。如果是 FF，则表示已经检查到了最后一个簇，在本应用实例里是簇 m，这时清理任务完成。此时，可以设定反馈一个“完成”的提示。

组件 2 利用组件 3 所提供的接口——比如目录访问接口等——来完成以上操作；而组件 3 对存储器等的操作则通过组件 4 提供的接口来完成。

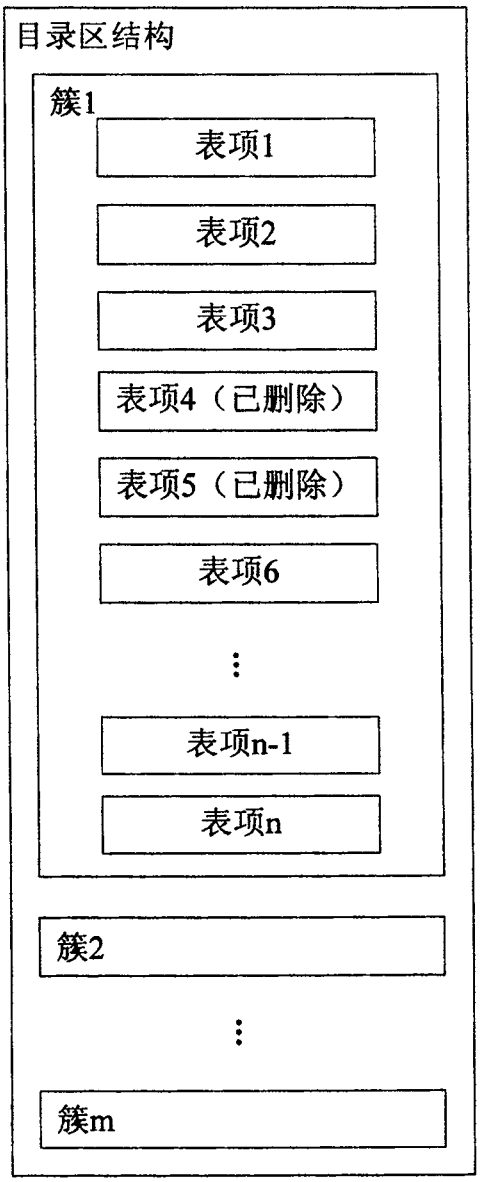


图 1

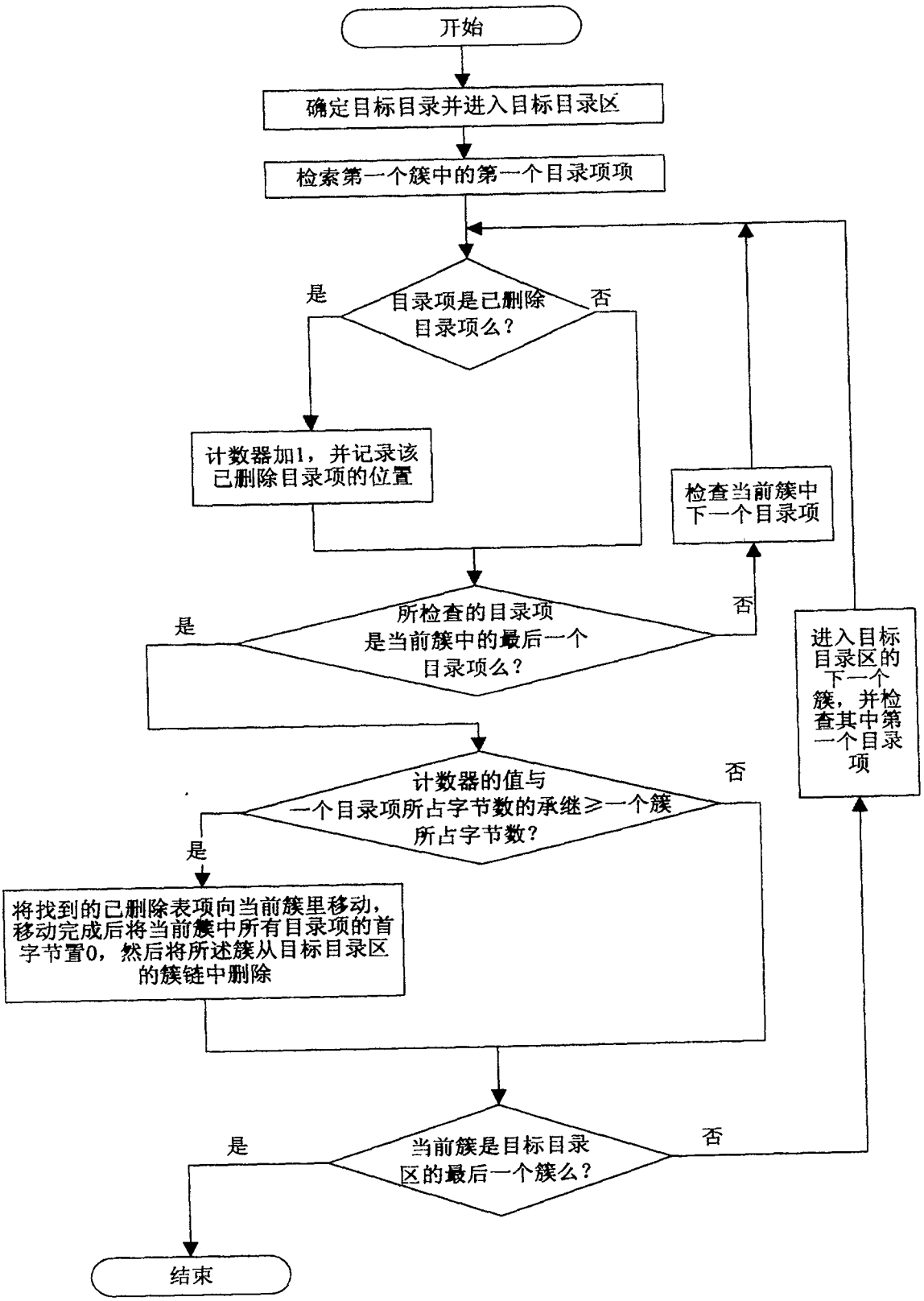

开始
确定目标目录并进入目标目录区
检索第一个簇中的第一个目录项项
目录项是已删除目录项么？
是
否
计数器加1，并记录该已删除目录项的位置
检查当前簇中下一个目录项
所检查的目录项是当前簇中的最后一个目录项么？
是
否
进入目标目录区的下一个簇，并检查其中第一个目录项
计数器的值与一个目录项所占字节数的承继≥一个簇所占字节数？
是
否
将找到的已删除表项向当前簇里移动，移动完成后将当前簇中所有目录项的首字节置0，然后将所述簇从目标目录区的簇链中删除
当前簇是目标目录区的最后一个簇么？
是
否
结束

图 2