



[12] 发明专利申请说明书

[21] 专利号 ZL 88100793

[51]Int.Cl⁵

G11B 27/10

[45]授权公告日 1993年3月3日

[24]颁证日 93.1.3

[21]申请号 88100793.5

[22]申请日 88.2.27

[30]优先权

[32]87.3.30 [33]US [31]032,210

[73]专利权人 国际商用机器公司

地 址 美国纽约州

[72]发明人 肯尼斯·活特·克里斯托福

巴里·阿兰·弗根包姆 金·基姆

道格拉斯·卡笨顿·洛夫

G06F 12/00

[74]专利代理机构 中国国际贸易促进委员会专利

代理部

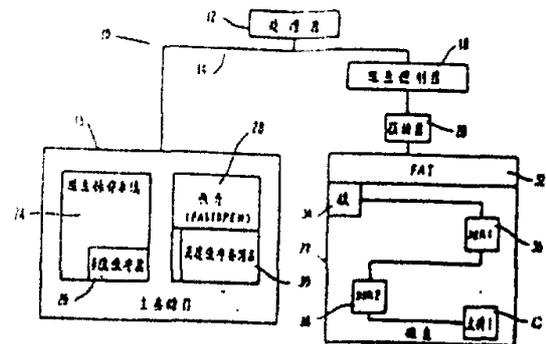
代理人 王以平

说明书页数: 附图页数:

[54]发明名称 快速开启由路径名识别的磁盘文件的方法

[57]摘要

一个数据处理系统具有以树型结构目录和文件的形式存贮在磁盘上的文件。该系统通过访问在主存贮器中某个驱动器高速缓冲存贮区而进行操作，以快速打开这些近期内已打开过或其部分路径信息有效的文件。高速缓冲存储区具有以树型结构链接的登记项，通过查寻这个树型结构，可以在打开文件过程中提供与否则必须由磁盘上得到信息相同信息。当高速缓冲存储区填满后，可用一个新项替换当前最少使用的项。



权 利 要 求 书

1. 操作具有第一磁盘驱动器的个人计算机系统的方法,该磁盘驱动器含有其上存有目录与文件的树型结构的第一磁盘,所述目录与所述文件被存储于所述磁盘上不同族中,每一目录包括识别存储于其中的文件和目录的目录登记项表,每一用于目录的目录登记项包括含有目录名及磁盘上存储这种目录的起始族信息,每一用于文件的目录登记项包括含有文件名及磁盘上存储所述文件的起始族的信息,所述数据处理系统进一步包括用于存储数据的主存储器,存储于所述存储器的所述数据以基本上比存储于所述磁盘上的数据所能存取的速度快的速度进行存取,所述方法的特征在于下列步骤:

(a)在所述主存储器内产生第一高速缓冲存储区,该第一高速缓冲存储区包括一些预定数量的高速缓冲存储区登记项,用以在所述磁盘驱动器内存储通向最近期用过的文件路径的历史,所述高速缓冲存储区登记项包括一用于每一文件的一个高速缓冲存储区登记项和用于这一文件的每一目录的一个高速缓冲存储区登记项,用于目录的高速缓冲存储区登记项包含用于这一目录的目录登记项,用于文件的高速缓冲存储区登记项包含用于这一文件的

目录登记项；

(b)响应打开嵌套于目录的一条路径上的给定文件的初始请求,在该路径上这一请求通过其文件名识别所述给定文件,通过其目录名识别该路径上的所述目录中的每一个,对于每一目录至少存取所述磁盘一次,对于所述给定文件至少存取所述磁盘一次,并从所述磁盘上读出对应于所述给定文件和通向所述给定文件的路径上之每一目录的那些目录登记项；

(c)将在步骤(b)读出的所述目录登记项存储于所述高速缓冲存储区内高速缓冲存储区登记项之中,并形成一個链,以产生通向所述给定文件的路径的历史；

(d)响应打开所述给定文件的后续请求,在所述链中搜索所述高速缓冲存储区登记项,将其中用于所述给定文件的所述目录登记项定位；以及

(e)利用在步骤(d)定位的所述目录登记项打开所述给定文件。

2. 根据权利要求1的方法,其特征进一步在于下列步骤：

(f)在步骤(a)中产生一个含有识别所述磁盘驱动器信息的高速缓冲存储区标题；

(g)在所述打开所述给定文件的后续请求中包含识别所述磁盘驱动器的信息；以及

所述步骤(d)包括在搜索所述高速缓冲存储区登记项之前,先

将在所述驱动器标题和所述后续请求中识别所述磁盘驱动器的信息进行比较。

3. 根据权利要求 1 的方法,其特征进一步在于,所述个人计算机系统包括多个磁盘驱动器,每一驱动器包括一个磁盘,各磁盘驱动器具有不同的驱动器 ID,包括:

在所述主存储器中产生多个高速缓冲存储区,包括在步骤(a)中所产生的所述高速缓冲存储区,对于每一磁盘驱动器有一个高速缓冲存储区,每一高速缓冲存储区包括一个高速缓冲存储区标题,其中存储有关的磁盘驱动器的驱动器 ID;以及

在打开文件的各请求中包括这些文件在其中定位的驱动器 ID;

根据来自前面步骤的各请求,搜索所述高速缓冲存储区标题,以定位包含用于欲被打开文件的登记项的高速缓冲存储区。

4. 根据权利要求 1 的方法,其特征在于,在所述步骤(a),提供一个其大小适于包含一预定数量登记项的高速缓冲存储区;

(f)根据打开多个不同文件的多个请求,用与所述不同文件有关的登记项填写所述高速缓冲存储区;

(g)在所述高速缓冲存储区存储文件使用历史,该历史指明哪一个文件是最近期使用过的;和

(h)然后,根据打开一附加文件的请求,删除那些与所述最近期使用过的文件相关的登记项,并加入与所述附加文件有关的登

记项。

5. 根据权利要求 1 的方法,其特征在于,根据打开一个由路径识别的第二给定文件的第三个请求,该路径包括某些与所述第一次提到的给定文件的路径中的某些目录相同的目录,在对应于这些公共目录的所述高速缓冲存储区中搜索所述登记项,然后从所述磁盘上读出这种进一步信息,以形成通往所述第二给定文件所必要的完整路径。

6. 根据权利要求 5 的方法,其特征进一步在于,从所述进一步信息中产生新登记项,并将其加给所述高速缓冲存储区,以在其中形成一个用于所述第二给定文件的登记项的完整路径。

7. 操作具有至少一个磁盘驱动器的个人计算机系统的方法,该磁盘驱动器包括一磁盘,第一文件在其上目录的树型结构中定位,所述目录和所述第一文件被存储在磁盘上不同位置,每一目录具有多个目录登记项,用以在磁盘上规定存储在这一目录中的每一文件和每一其它目录的名称和位置,在包含所述第一文件名和位置的所述目录之一中有一个第一目录登记项,所述个人计算机系统进一步包括一主存储器,在主存储器中,存储于其中的数据可以以基本比在所述磁盘中数据所能存取的速度快的速度存取,其特征在于该方法包括下列步骤:

分配所述存储器的一部分用以存储多个记录;

请求打开所述第一文件,在通过经所述树型结构通向所述第

一目录登记项的一条第一路径上,由规定其文件名和各目录名称来识别所述第一文件;

通过包括对于所述第一路径上的每一目录至少存取所述磁盘一次的步骤打开所述第一文件,并搜索所述第一路径中的每一目录,以便定位所述第一目录登记项;和

在所述存储器中所述记录内存储用于所述第一路径上所述目录的目录登记项和用于所述第一文件的所述第一目录登记项,这些目录登记项被读出,而在前一步骤中它们存取所述磁盘。

8. 根据权利要求 7 的方法,其特征进一步在于,通过搜索含有存储于其中的所述目录登记项的所述记录,后续地打开所述第一文件,并将所述第一目录登记项定位于所述第一记录中,从而消除为从中获得这一第一目录登记项而存取所述磁盘驱动器的必要。

9. 根据权利要求 8 的方法,其特征在于,

后续地打开存储在所述磁盘上所述树型结构中第二文件,其中所述第二文件具有通向该文件的第二路径,该路径包括一个或更多在所述第一路径中含有的目录,在所述第二路径上所述目录之一包含带有所述第二文件名及位置的所述第二目录登记项;

首先在用于与两个所述路径上的所述目录相对应的记录的所述存储器中,搜索所述记录链;及

访问所述磁盘,以使将在所述第二路径中的任何剩余目录和所述第二目录登记项定位。

10. 根据权利要求 9 的方法,其特征在于,在所述记录中存储对应于所述第二文件和任何在所述第二路径上不包括在所述第一路径上的目录的那些目录登记项。

快速开启由路径名识别的磁盘文件的方法

本发明涉及有关数据处理的领域,尤其是涉及一种快速打开由路径名识别的磁盘文件的方法。

本发明旨在对众所周知并已商业化的这类系统进行改进,例如 IBM 个人计算机,在这类系统中,硬件是在磁盘操作系统(DOS)的管理下操作的,并且磁盘文件采用树型结构目录和路径名存贮在硬磁盘中,信息按照柱面和扇区的预定格式存贮在磁盘上,每个扇区包含预定的字节数。为了访问所期望的扇区,磁头必须首先移至包含该扇区的柱面上,转动磁盘,使之经过磁头,直到磁头到达所需的扇区,才能读出该扇区的内容并将它放在缓冲区中。检查为访问磁盘上的数据所需的总时间可以看出,主要延时发生在磁头实际移动的过程中,在某一应用需要运行大量实际 I/D 设备的情况下,希望尽可能地减少磁头移动的次数。

文件按一簇或多簇扇区存贮在磁盘上,每个簇包含预定的扇区数。每个簇有一个唯一不同的起始地址。磁盘上文件的位置是借助于文件分配表(FAT)记录的,文件分配表本身存贮在该磁盘上。FAT 的每个位置与不同的簇相关并且包含一个登记项,该登记项用来指示不存在其它簇与某一文件相关或指向该文件的下一个簇号。

短的文件包含在一个簇中。长文件则包含在若干链接在一起的簇中。

文件通过使用树型结构目录来确定位置。每个磁盘包含一个根目录,若干子目录和大量文件。一个给定文件可以位于经过根目录和若干子目录的路径的尾部。每个目录包含一个用于附加目录和文件的登记项。一个特定文件可以通过定义驱动器、路径和文件名来识别。例如:*C:/DIR1/DIR2/FILE1* 识别列目录 *DIR2* 中的文件名 *FILE1*, *DIR2* 是 *DIR1* 的子目录,并列在其中,而 *DIR1* 又是驱动器 *C* 的根目录的一个子目录,也列在其中。

当打开一个文件时,必须访问驱动器并搜寻这个路径上指定的所有目录,确定包含该文件名登记项的目录。再在这个目录里,搜寻所有的登记项和文件名直到找到所需要的文件为止。如果某个文件以前没有打开过,则没有登记项,因此在使用该文件前,必须设置登记项。如果该文件以前打开过,则含有该文件名的目录中的登记项又包含着一个登记项,该登记项是指向 *FAT*(对应着该文件起始位置的簇)的索引。在这种打开过程中,为访问根目录、每个子目录和搜寻一长例文件名,必须进行物理的 *I/O* 操作。在某些应用中,在运行给定程序的过程中,相同的文件多次打开,并且每次打开文件都包括大量的物理 *I/O* 操作,因此要损失相当多的时间。本发明旨在改进现有开启文件的方法,即提供一种快速打开文件的方法并减少与开启文件过程相关的物理 *I/O* 操作的次数。

本发明的目的之一是提供一种在某文件最初已被至少打开一次之后,快速打开该文件的方法。

其另一目的是提供一种在高性能个人计算机系统中快速打开文

件的方法,其具体作法是减少与开启文件过程有关的物理 I/O 操作的次数。

其目的之三是通过下述方法减少打开文件所需的时间,即,一旦该文件以前已经打开过,则去掉按通常为确定目录中的文件登记项而搜寻整个磁盘驱动器的目录以及不同文件名的过程,从而减少打开文件的时间。

本发明的目的之四是提供快速再次打开那些近期已访问过的文件以及快速再次打开那些频繁开、闭的嵌套文件的方法。

简言之,做到这一点的方法是在主存贮器的高速缓冲存储区中建立并保持文件使用的历史。每当访问这一个目录或文件时,先检查其历史,如果打开文件所需的信息在高速缓冲存贮区中,则可直接使用,无需对可能涉及物理 I/O 操作的任一目录或文件名进行树型搜索。通过保持主存贮器的高速缓冲存储区中的内容,这类信息就可以按主存贮器的高速度进行访问,而不涉及与磁盘的物理 I/O 操作有关的较低速度。

具体的说,本发明旨在提供一种操作一个至少具有一个磁盘驱动器的数据处理系统的方法,该磁盘驱动器包含一存储有树型结构目录和文件的磁盘,各文件由路径和由文件名识别,所述路径包括许多目录,其中包含有一个具有一登记项的目录,登记项中包含所述文件名和指示该文件存储在所述磁盘上何处的信息。

①提供一用于存储信息的高速缓冲存储区,所述高速缓冲存储区中的所述信息可以以一种比存储在所述磁盘中的信息所能存取的速度快的速度存取;

②响应打开给定文件的第一个请求,搜寻所述存储在所述磁盘中的许多目录,以便从中读出所述给定文件的文件名和指示所述给定文件位于所述磁盘上何处的信息;

③把许多链接的登记项存储在所述高速缓冲存储区中,这些链接的登记项包括一个与在通向所述给定文件的路径中的各个所述目录相应的登记项及一个包含所述给定文件的文件名和所述指示该给定文件位于所述磁盘何处的信息的登记项;

④响应打开所述给定文件的第二个请求,搜寻所述许多链接的登记项,以决定所述给定文件的文件名以及所述指示该给定文件存储于所述磁盘何处的信息的位置;

⑤用步骤④中找出的所述信息打开所述给定文件。

其中步骤①中还包括提供许多逐个与所述磁盘驱动器相对应的高速缓冲存储区,它们各包括一驱动器高速缓冲存储区标题,用以识别每一个不同的所述磁盘驱动器。

步骤④中包括首先至少存取一个所述驱动器高速缓冲存储区标题,以确定包括许多链接的登记项的高速缓冲存储区的位置。

所述高速缓冲存储区的大小能包含预定数量的登记项。

本发明的方法还包括下列步骤:

⑥响应许多打开各个不同文件的请求,用与所述不同文件相关的登记项填充所述高速缓冲存储区;

⑦把文件使用历史存入所述高速缓冲存储区,该文件使用历史指示出哪个文件是当前最少使用的文件;

⑧响应打开一附加文件的请求,删去与当前最少使用的文件有

关的那些登记项,加上与所述附加文件相关的登记项。

步骤⑥中还包括存储登记项中的指针,以产生一个从当前最常使用的文件到最少使用的文件顺序排列的登记项链。

本发明的方法还进一步包括下列步骤:响应打开第二给定文件的第三请求,在高速缓冲存储区中搜寻与公共目录相应的登记项,然后从磁盘中读出必要的进一步信息,以形成通往所述第二给定文件的完整路径,该第二给定文件是通过包括与所述第一次提到的给定文件的路径中某些目录相同的一些目录的路径来识别的。根据上述进一步信息产生的新的登记项,把这些新的登记项加到所述高速存储区中形成完整的用于第二给定文件的登记项路径。

通过下面参照附图所进行的描述,本发明的其它目的及优点将是显而易见的,其中:

图 1 是一般性说明本发明的实现上述方法的环境及现有技术相互关系的简图。

图 2 是响应用户请求,按本发明所做的一般步骤流程图。

图 3 是(一个流程图)用于选择各个具体过程的主过程流程图。

图 4 是查寻过程流程图。

图 5 是插入过程的流程图。

图 6 是清除请求过程的流程图。

图 7 是删除过程的流程图。

图 8 是更新过程的流程图。

图 9 是说明 DOS 如何调用本发明某些过程的流程图。

现在参考附图,首先看图 1。数据处理系统 10(例如 IBM AT

个人计算机)包括,通过总线 14 与主存储器 16 和磁盘控制器 18 相连的处理器 12。磁盘控制器又依次与包括一个磁盘 22 的磁盘驱动器 20 相连,磁盘具有记录信息位的常规存储介质。在系统 10 操作期间,DOS 过程 24 装入或驻留在主存储器 16 中,而它主要用于控制对磁盘 22 上的信息的访问。DOS 与系统缓冲器 26 相关联,即通过系统缓冲器使信息在磁盘 22 和主存储器 16 之间按通常方式进行传送。

还有一组叫作 FASTOPEN28 的过程也装入并驻留在主存储器 16 中,该过程与 FASTOPEN 高速缓冲存储区 30 的操作有关,其细节将在下面描述。如果该系统中的驱动器多于一个,是将有相应数目的高速缓冲存储区 30,即每个存储区 30 对应于一个驱动器。磁盘 22 本身已格式化,以使其包含一个文件分配表 32(FAT)和根目录 34。图 1 中还给出一个作为示例的文件结构,在该结构中,名为 FILE1 的文件 40 位于名为 DIR2 的子目录 38 中,DIR2 本身位于名为 DIR1 的子目录 36 中,而 DIR1 又是根目录 34 的子目录。如果没有装入过程 FASTOPEN28,则该系统按常规已知的方式操作。FASTOPEN28 的装入通过下述过程实现,即对下面将介绍的各变量进行初始化,然后使其终止,但驻留在主存储器 16 中,以便可根据需要进行调用。

如上所述,在该系统中每个磁盘驱动器对应有一个快速缓冲存储区 30。每个快速缓冲存储区包含一块称为驱动器高速缓冲存储区标题的信息。这个标题由若干不同字段组成,其意义如表 1 所示。

表 1 驱动器高速缓冲存储区标题

字 段	意义
1	这个驱动器的 <i>LRU</i> 链的标题
2	至 <i>LRU</i> 最后一个登记项的偏移量
3	指向位于登记项链中第一个子项的指针
4	指向下一个驱动器高速缓冲存储区标题的指针
5	驱动器 <i>ID</i>

在每个高速缓冲存储区 30 中设有若干目录和文件登记项,每个登记项包含如表 2 所列的若干信息字段。这些登记项组成一个链型数据结构,这个结构将在下面详细描述。登记项的链连是由适当的指针把这些登记项依次连接起来而达到的。简单地说,这些登记项由表 2 中列出的指针链连起来。高速缓冲存储区 30 的尺寸将根据所希望的登记项数目或由用户所选的登记项数目来确定,上述所希望的登记项个数可通过系统省缺值选定,通常,由用户所选的数目应大于在驱动器中经登记项的最深嵌套。

表 2 登记项

字 段	意义
1	指向下一个 <i>LRU</i> 登记项的指针
2	指向下一个 <i>MRU</i> 登记项的指针
3	指向子项的指针
4	指向兄弟项的指针
5	指向前一个结点的指针
6	目录/文件信息

较佳的作法是以下述方式将登记项维护在一个链中,这个方式

是当高速缓冲存储区 30 充满登记项且需加入一个新项时,这个新登记项将取代高速缓冲存储区中当前最少使用项 (LRU)。因此,根据 LRU 和 MRU(当前最频繁使用的项)的概念,用字段 1 和字段 2 将各登记项链接起来。登记项是针对每个已打开的文件以及每个含在通向该文件的路径中的子目录增设的。在同一子目录中的两个文件、两个目录或一个文件、一个目录认为是处于同一层,且称之为“兄弟”,其他子目录下的任一子目录或文件的登记项称为“子”。为了搜寻的目的,在高速缓冲存储区中的各登记项通过指针字段 3—5 以树型结构排列。字段 6 包含与存储在磁盘介质中的普通目录项有关的信息,除非该信息已插入到名字项的字段 6 中并因此驻留在主存储器里。因此,当需要打开一个在高速缓冲存储区 30 中存有相应登记项的文件时,这个信息能被快速存取。关于一个文件的信息包括:文件名和扩展名、属性、最后写入时间、最后写入日期、文件中第一个簇号及文件尺寸。

参考图 2,当一个应用或用户程序想打开某文件时,就要采用这个已知的用于打开文件 50 的“DOS 调用中断 21Hex”。该调用进入 DOS24,并且如前所指装入 FASTOPEN,以便可截取这个中断并对其发生作用。该过程的第一次判定是确定是否已装入 FASTOPEN,它由步骤 52 实现。如果没有装入,则流程转向现有技术的步骤 54,即以通常方式访问磁盘来确定每个子目录和文件的位置,此后由步骤 56 返回用户或应用。如果已装入 FASTOPEN,则步骤 58 按下面将详细描述的方式查寻请求。如果该过程出现错误,则从步骤 60 分支到步骤 54 中,并且该过程按照现有技术的过程进行处理。如果没

有出现错误,则步骤 62 确定该路径是部分被找到,或相反地已全部被找到,如果不是部分找到,即已找出整个路径以及文件名,则得到和步骤 54 所提供的相同信息,经由步骤 56 返回至用户。如果仅找到部分路径,那么转入步骤 64,从已找出的信息点出发再继续访问磁盘以确定余下部分的路径和文件。在步骤 66 中插入寻找到的各个部分,而步骤 68 使该过程重复从步骤 64 开始进行,直至到达该路径的尾部或找到文件名为止,同时将相应的信息返送给用户。

参考图 3,这个过程的核心部分表示为“主(main)”,它用于确定应使用四个子过程中的哪一个过程。这四个过程分别是“查寻”,“插入”,“删除”,和“更新”。这些不同的过程可以这样来调用,例如,通过把与之相对应的不同的数值插入处理器寄存器中的一个来实现。在任何情况下,步骤 70 判定是否是查寻操作,同样地步骤 72、74 和 76 分别判定是否是插入、删除或更新过程。如果这些过程均未调用,则返送回一个错误信息。下面根据各过程的特点,分别介绍这些过程。

参考图 4,根据查寻过程,步骤 78 首先确定与所申请的驱动器相关的高速缓冲存储区 30。然后步骤 80 从预 LRU(pre-LRU)堆栈中设置该 LRU 链。这堆栈是一个逻辑堆栈,用于在维护 LRU 链进保持这个树型结构子目录。下一步,步骤 82 形成一个回路的起始段,在这个回路中,现行结点是一具指针,它最初被设置指向该驱动器的首部,然后被设置指向现行结点的登记项。现行“兄弟”被设为零且开始查寻目录或文件名。遇到零值 ASC II 码就确定为路径的结尾或文件名,由步骤 84 判定这个过程。如果没有遇到零值 ASC 码(这是在一次查寻刚刚开始时的一般情况),则执行步骤 86,使用指向现行

登记项的子项的指针,从而找出现行结点下的一个项。步骤 88 检查是否存在其它子项,即,该登记项在该序列中是否为最后一个子项?这个事实将由在子字段中的-1 来指示。然后将期望的名字与该子项名字比较。如果由步骤 90 进行比较的结果成功,则步骤 92 生成一个 *Pre-LRU* 堆栈,将现行结点地址保存在该堆栈中并分支返回步骤 82。这个过程连续循环进行直到找出文件名,此时由步骤 84 得到的“yes”或肯定结果则分支到步骤 94 并将已找到的记录或信息返送回 *DOS*。如果步骤 90 给出一个否定决断,则步骤 98 查寻指向“兄弟”字段的指针并在步骤 100 开始一个经步骤 96、90 和 98 的循环过程,该循环直至无剩余“兄弟”时结束。此时该步骤 100 开始的循环过程经步骤 94 返回 *DOS*。当返回 *DOS* 后,*FASTOPEN* 将设置该指针指向从 *DOS* 提供的路径/文件名字符串中找出的这个目录或文件名。*DOS* 又将依次查寻指针以便确定是否已找出整个路径,或者是找出部分路径或者什么也未找到。当存在较多“兄弟”时,步骤 96 设置现行结点指针指向现行的“兄弟”。

参考图 5,插入过程的目的是:基于现行结点和现行的“兄弟”信息,将目录或文件信息作为一个登记项插入到高速缓冲存储区中。步骤 102 由输入产生一个名字记录项,步骤 104 得到了一个可用项。在高速缓冲存储区未装满之前,一个新项可以简单地插入到任一位置,即登记项是自由的。一旦高速缓冲存储区充满之后,则根据 *LRU* 表得到的当前最少使用的项就成为要由新项替代的下一个可用项。步骤 106 把预 *LNKR* 堆栈清理干净。如果现行“兄弟”指针等于零,则经步骤 110 将一个新项作为现行结点的一个子项插入,步骤 112

在预 LRU 堆栈中记录该结点。如果步骤 108 中的现行“兄弟”不为零,则步骤 114 将一个新项作为现行结点的“兄弟”插入。

参考图 6、图 7,当某应用或用户请求删除一个特定文件时,则该过程不仅按通常方法删除磁盘中的该文件,而且删除高速缓冲存储区中与该文件有关的各登记项。因此,步骤 116 响应用户的请求,流程经 DOS24 由步骤 118 确定是否已装入 FASTOPEN。如果没有装入,则步骤 122 访问磁盘并删除该文件,删除后经步骤 124 返回用户。如果已装入 FASTOPEN,则发出删除请求 120。为响应这个请求,步骤 126 查寻高速缓冲存储区以确定是否存在相应的登记项。如果在步骤 128 没有找到该登记项,则简单地结束这个过程。如果找到该登记项,则通过步骤 130 将它从树型结构 LRU 链中移出并通过步骤 132 把此特定的项放在 LRU 链的顶部,作为可以使用的新项。首先删除一个文件项,然后删除那些不包含任何子项或兄弟项指针的任何“父”目录。

参考图 8、图 9,这里有某些位于 FASTOPEN 过程中并可由 DOS 直接调用的操作,用于完成某些附属于 DOS 所提供的功能的操作,DOS 的请求附属于下述功能:即关闭文件、向文件里写一个零字节或产生一个节点,步骤 142、144 和 146 分别确定是否请求这些功能,然后分支到更新步骤 134—140。由步骤 136—140 的肯定结果分别导致更新一个目录项,更新与某文件有关的簇号以及删除某个登记项。

在上述的讨论中,只谈到预 LRU 堆栈,现在介绍与 LRU 堆栈有关的内容,这两种堆栈均涉及使用位于登记项中的 LRU 指针和

MRU 指针。当该过程走过一条路径时,从根目录开始,经各子目录直至到达文件名,这对按照各结点的访问顺序生成预 LRU 堆栈。因此,以由 C:/DIR1/DIR2/FILE1 访问或指定文件的例子中,各项的次序是根目录、DIR1、DIR2 和 FILE1:这种排列产生一个问题,即根目录应是当前最少使用的项,而为了重进行置换就应删除该项,那么要存取其余的各登记项就无法进行了,一旦路径建立起来,通过使用 Pre-LRU 堆栈和重新将各项排序,从而使生成的 LRU 堆栈以文件名作为当前最少使用项。因此当要进行项的替换时,删掉的将是文件名,而不是象前面那样删掉任一双亲目录了。

显然,本领域的技术人员无需违反由权利要求书所规定的本发明的基本精神和范围,就可以进行其它变更和改进。

图.1

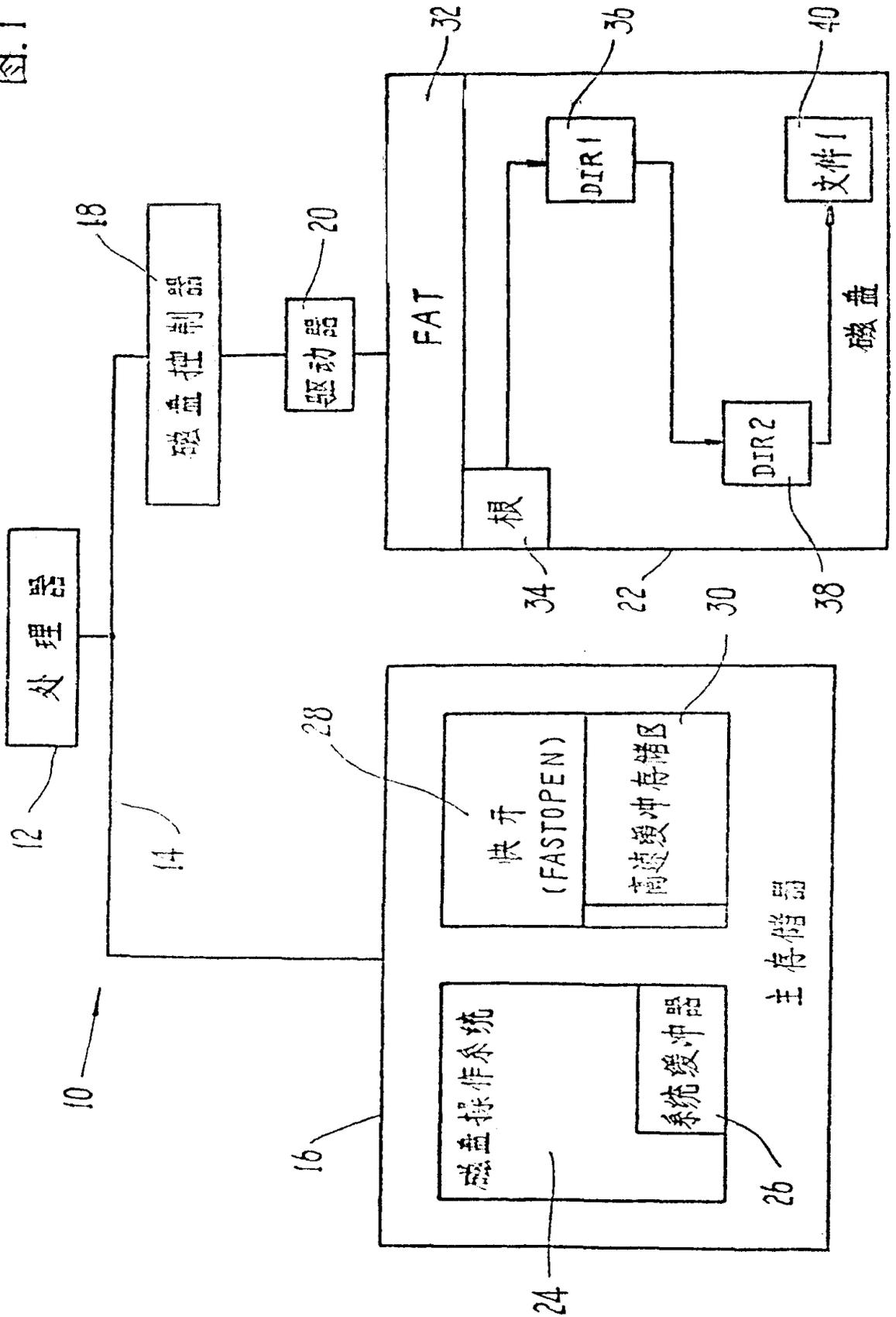


图. 2

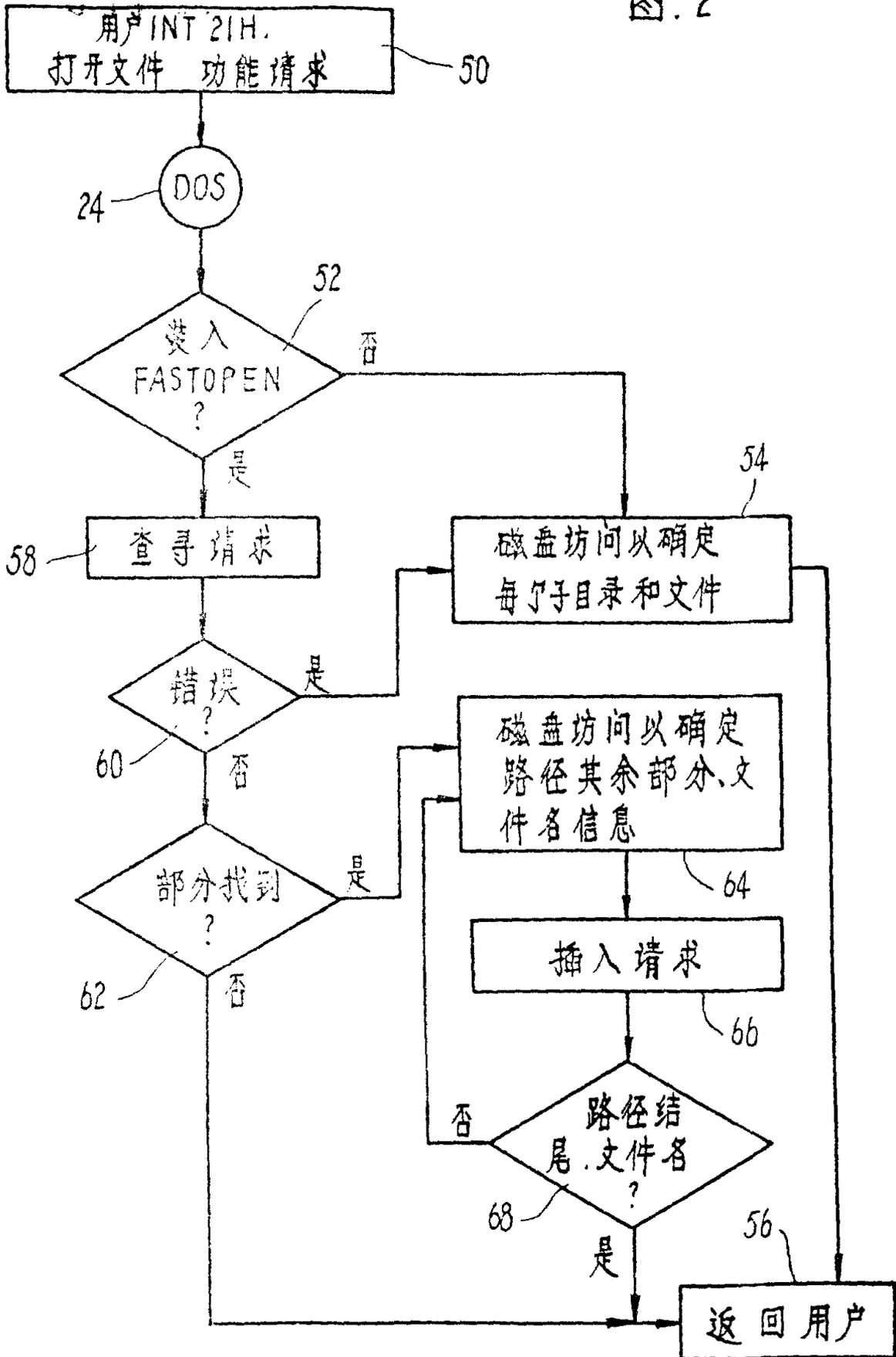


图. 3

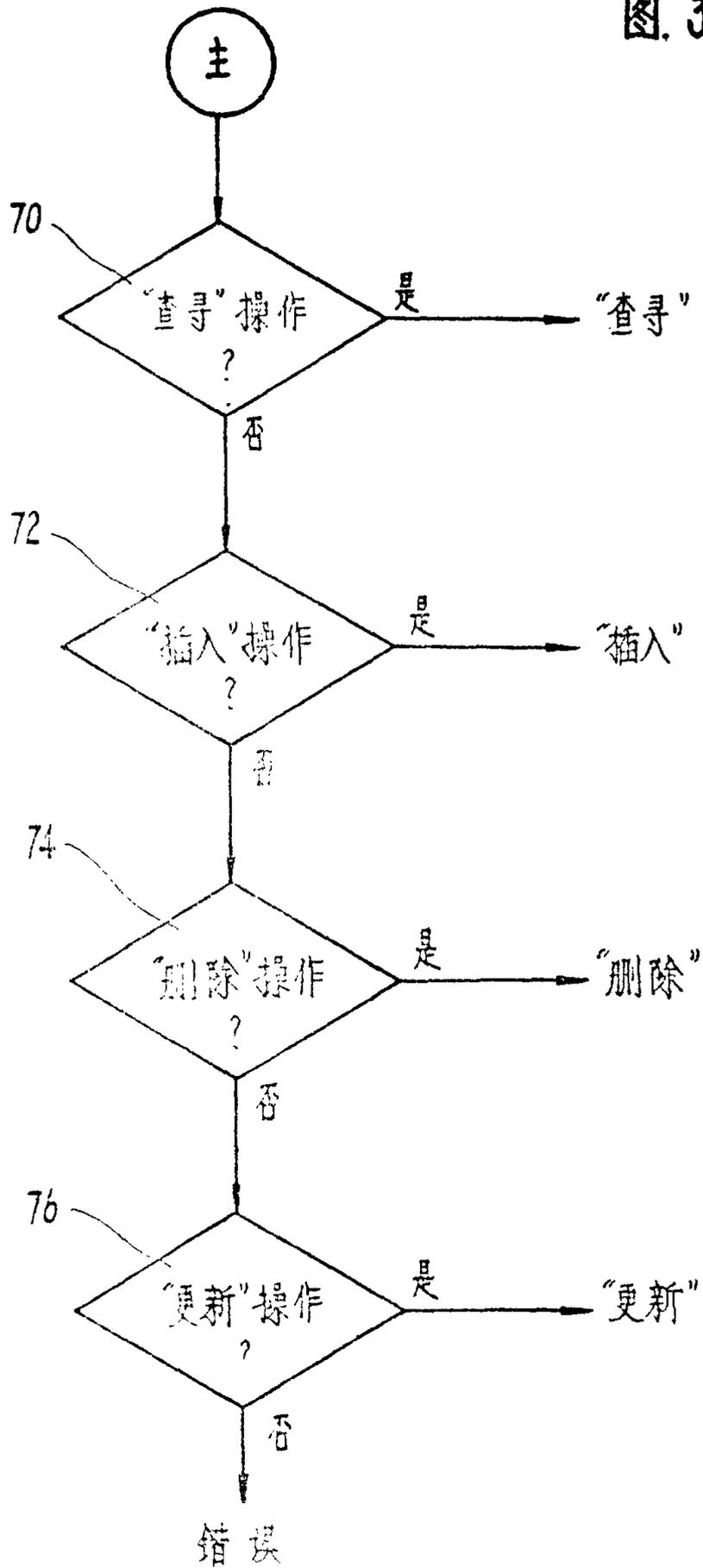


图. 4

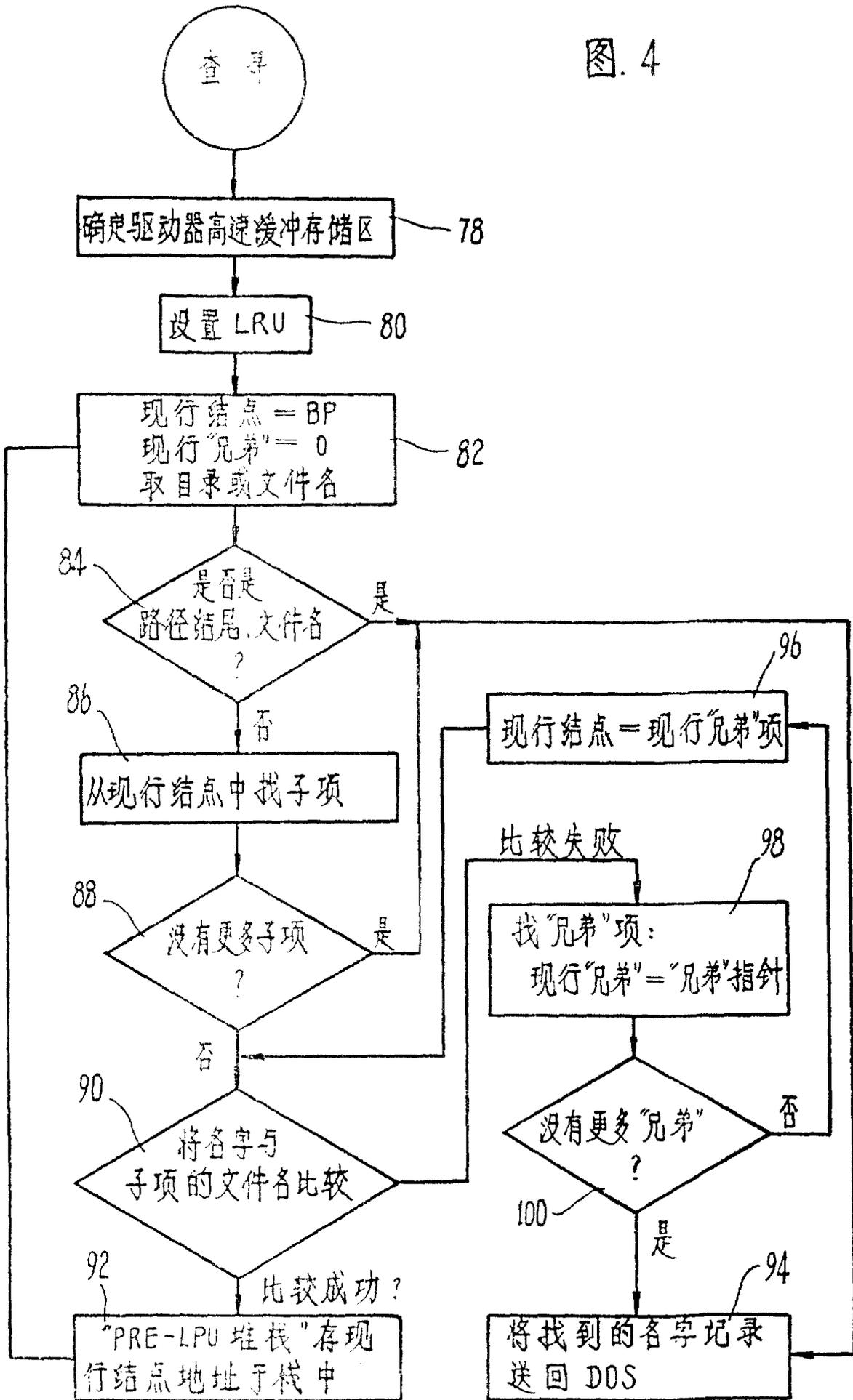
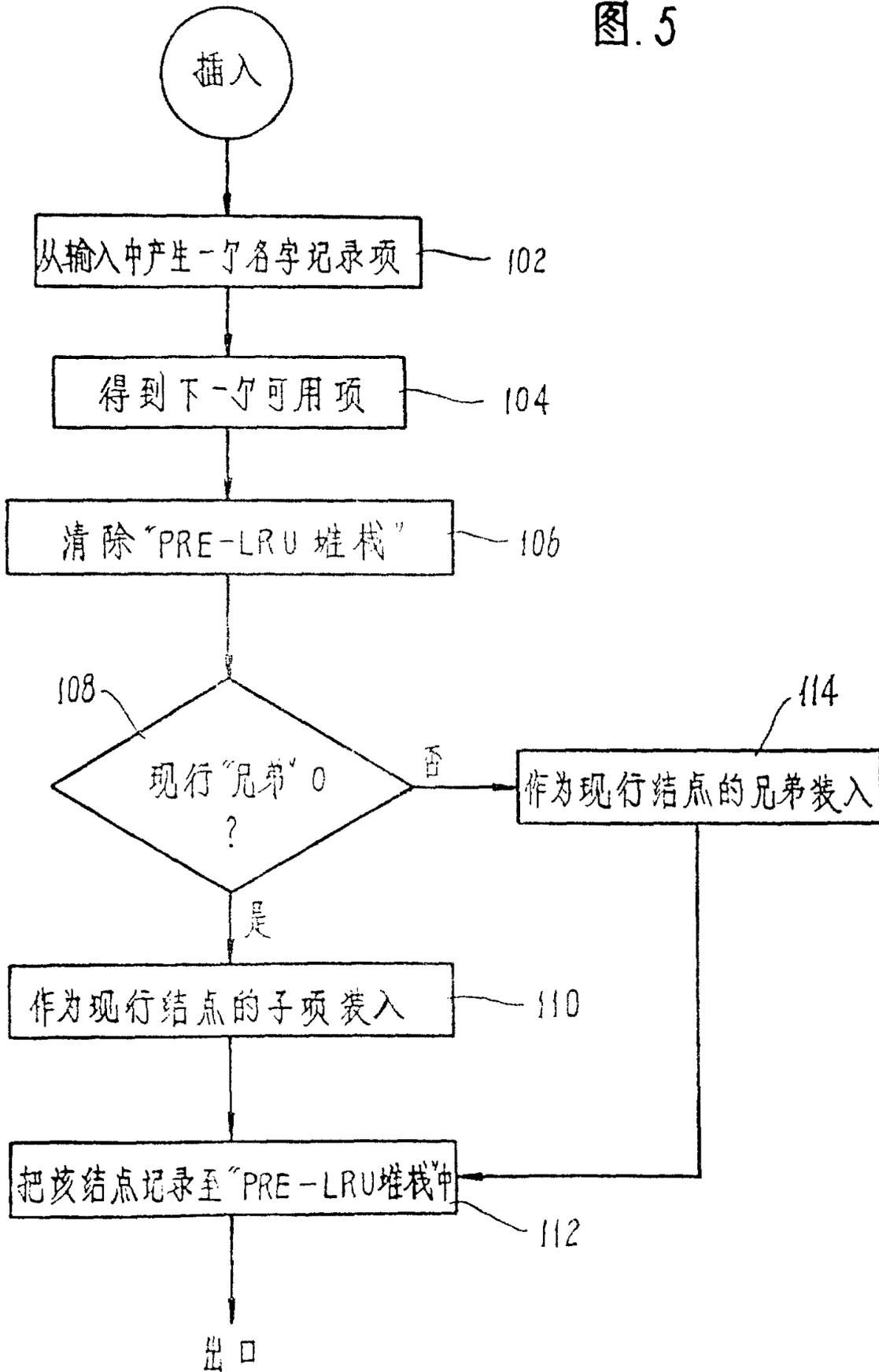


图. 5



用户的 INT21H 删除文件功能请求

图. 6

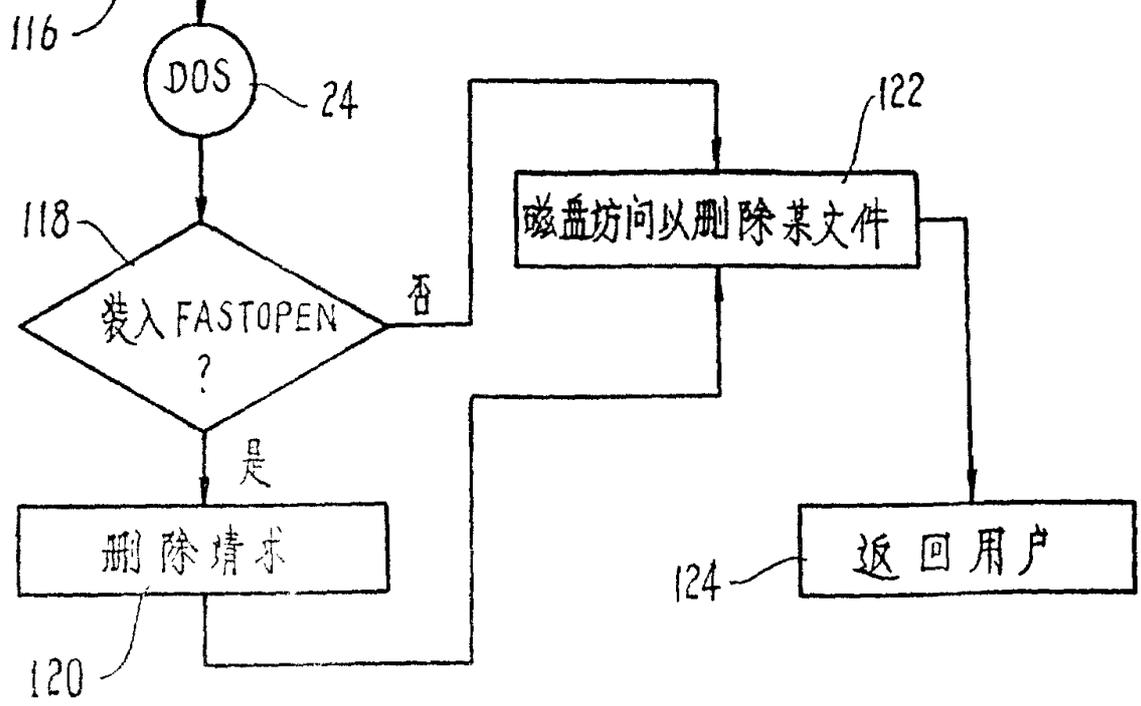


图. 7

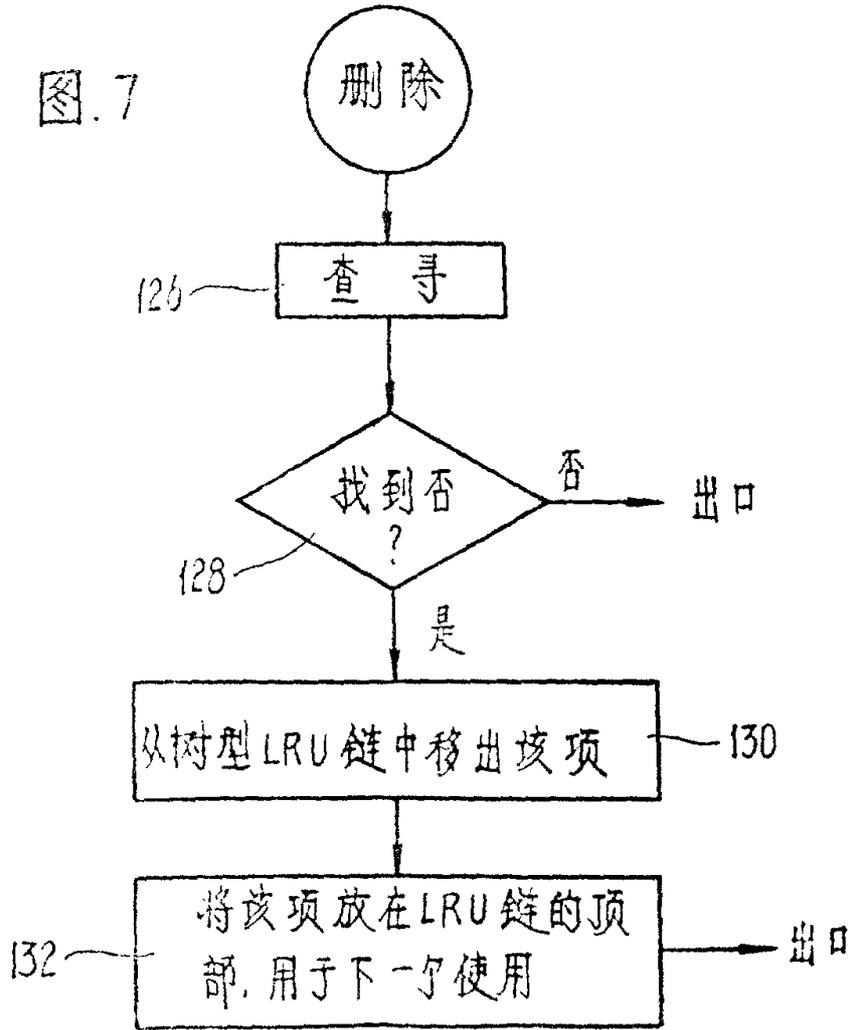


图. 8

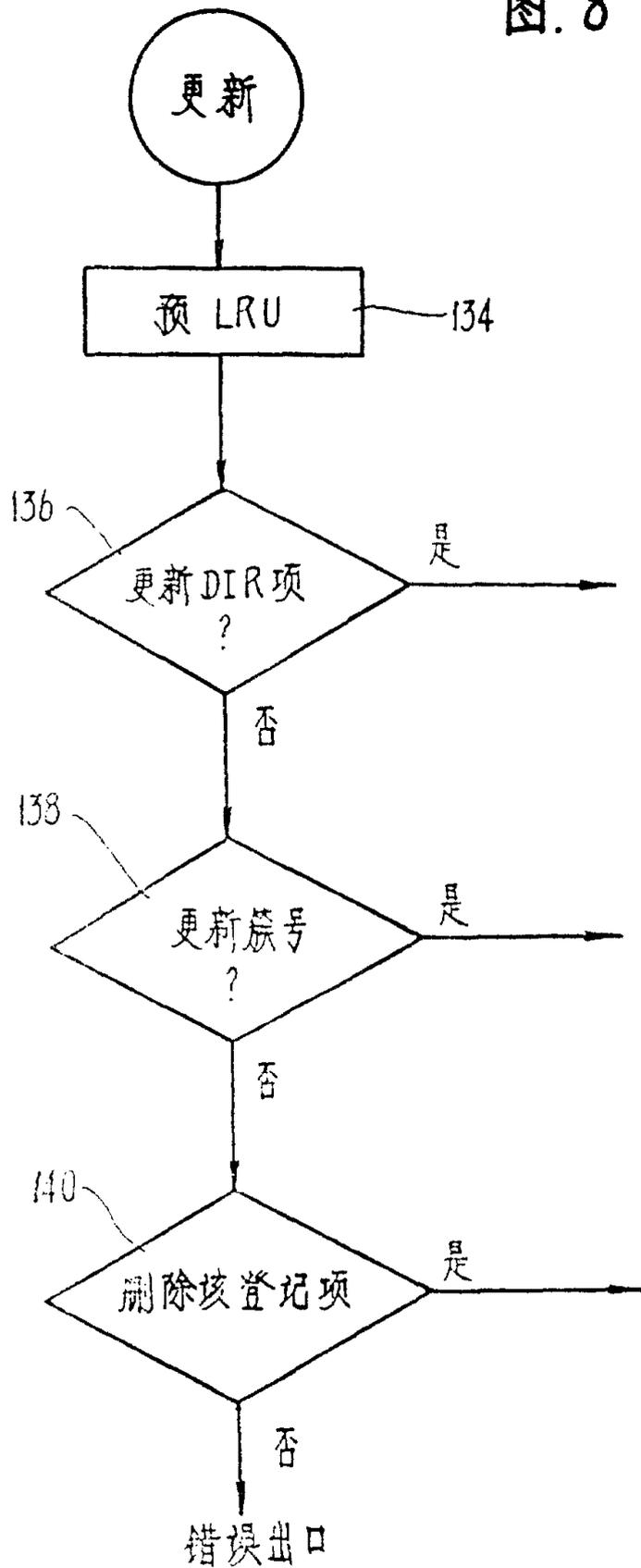


图. 9

