



[12] 发明专利申请公开说明书

[21] 申请号 200510119239.1

[43] 公开日 2006年6月7日

[11] 公开号 CN 1783082A

[22] 申请日 2005.10.28

[21] 申请号 200510119239.1

[30] 优先权

[32] 2004.11.30 [33] US [31] 11/000,430

[71] 申请人 微软公司

地址 美国华盛顿州

[72] 发明人 N·R·克里斯琴森

R·S·蒂恩德 S·C·哈弗瓦拉

[74] 专利代理机构 上海专利商标事务所有限公司

代理人 张政权

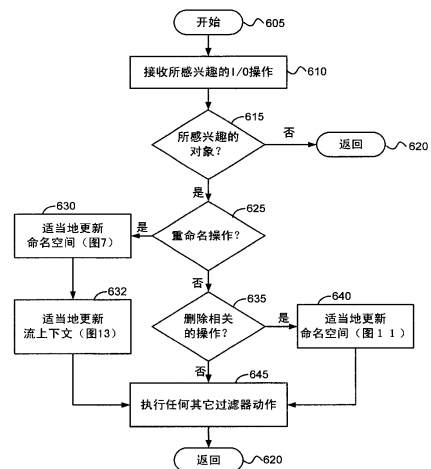
权利要求书 3 页 说明书 17 页 附图 15 页

[54] 发明名称

用于与文件系统保持命名空间一致性的方法和系统

[57] 摘要

一种用于在由文件系统维护的选择的对象及与之关联的过滤器之间保持命名空间一致性的方法和系统。过滤器监控选定类型的请求(或与之关联的操作)并确定对象是否在与过滤器关联的命名空间内。与过滤器关联的命名空间基于对象的更改被更新。



1. 一计算机可读介质，具有计算机可执行指令，包括：  
接收访问文件系统对象的请求；  
确定所述对象在与过滤器关联的命名空间内；以及  
基于对于所述对象的更改更新与所述过滤器关联的命名空间。
2. 如权利要求 1 所述的计算机可读介质，其特征在于，所述访问对象的请求包括重命名操作。
3. 如权利要求 2 所述的计算机可读介质，其特征在于，所述重命名操作导致对象移动至文件系统内的不同位置。
4. 如权利要求 2 所述的计算机可读介质，其特征在于，所述重命名操作导致与对象关联的名称更改。
5. 如权利要求 1 所述的计算机可读介质，其特征在于，所述访问对象的请求包括删除操作。
6. 如权利要求 5 所述的计算机可读介质，其特征在于，所述删除操作包括设定标记以指示在与对象关联的所有句柄关闭后对象被删除。
7. 如权利要求 6 所述的计算机可读介质，其特征在于，还包括清除标记以指示对象不被删除。
8. 如权利要求 1 所述的计算机可读介质，其特征在于，所述访问对象的请求包括清除操作。
9. 如权利要求 8 所述的计算机可读介质，其特征在于，每次进程关闭对象的句柄时，提交清除操作。
10. 如权利要求 1 所述的计算机可读介质，其特征在于，所述对象是否在与过滤器关联的命名空间的确定包括在由过滤器维护的数据结构内搜索对象名。
11. 如权利要求 10 所述的计算机可读介质，其特征在于，所述数据结构分层地排列。
12. 如权利要求 10 所述的计算机可读介质，其特征在于，所述数据结构包括每个对象的节点，所述对象具有与之关联的策略。
13. 如权利要求 12 所述的计算机可读介质，其特征在于，所述数据结构还包括每个对象的每个祖先的节点，所述对象具有与之关联的策略。

14. 如权利要求 13 所述的计算机可读介质，其特征在于，所述祖先包括目录，对象驻留在所述目录内。
15. 如权利要求 1 所述的计算机可读介质，其特征在于，所述对象包括文件系统的目录。
16. 如权利要求 1 所述的计算机可读介质，其特征在于，所述对象包括文件系统的文件。
17. 如权利要求 1 所述的计算机可读介质，其特征在于，所述对象的更改包括对象名的更改。
18. 如权利要求 1 所述的计算机可读介质，其特征在于，所述对象的更改包括文件系统内对象位置的更改。
19. 在一计算环境内，一方法，包括：  
打开文件系统的对象的句柄；  
确定所述对象是否与过滤器命名空间相关；以及  
如果对象与过滤器命名空间相关，则基于对象的更改更新过滤器命名空间。
20. 如权利要求 19 所述的方法，其特征在于，所述打开文件系统的对象的句柄是创建的一部分。
21. 如权利要求 20 所述的方法，其特征在于，所述创建包括打开操作。
22. 如权利要求 20 所述的方法，其特征在于，所述创建包括为关闭操作后删除而打开，其中在对象的所有句柄被关闭后，对象被删除。
23. 如权利要求 20 所述的方法，其特征在于，所述创建导致对象在文件系统上被创建。
24. 如权利要求 19 所述的方法，其特征在于，所述过滤器包括在所述请求发送至文件系统之前，有机会检测涉及发送至文件系统的请求的数据的进程。
25. 如权利要求 19 所述的方法，其特征在于，所述过滤器包括在所述结果数据返回发出所述请求的请求者之前，有机会检测涉及发送至文件系统的请求的结果数据的进程。
26. 如权利要求 19 所述的方法，其特征在于，所述过滤器登记以接收对于某些与卷相关的操作的数据。
27. 如权利要求 19 所述的方法，其特征在于，所述过滤器命名空间包括表示策略对其存在的对象的数据。

- 
28. 如权利要求 27 所述的方法, 其特征在于, 所述过滤器命名空间还包括标识对象的每个祖先的数据。
29. 如权利要求 28 所述的方法, 其特征在于, 所述数据存储在被分层排列的节点中。
30. 如权利要求 19 所述的方法, 其特征在于, 所述过滤器包括内核模式进程。
31. 如权利要求 19 所述的方法, 其特征在于, 所述对象的更改包括删除对象。
32. 如权利要求 19 所述的方法, 其特征在于, 还包括确定更改是否成功。
33. 如权利要求 19 所述的方法, 其特征在于, 还包括更改对象的文件系统名。
34. 如权利要求 19 所述的方法, 其特征在于, 还包括更改对象删除配置。
35. 一用于同步命名空间的装置, 包括:  
被安排以存储对象的文件系统, 所述文件系统具有第一命名空间;  
被安排以请求访问所述对象的进程; 以及  
被安排以检测所述文件系统和所述进程之间选择的通信并对于任何影响第二命名空间的第一命名空间的更改更新第二命名空间的过滤器。
36. 如权利要求 35 所述的装置, 其特征在于, 如果对象包括在第二命名空间中且对象被删除或重命名, 则第一命名空间的更改影响第二命名空间。
37. 如权利要求 36 所述的装置, 其特征在于, 所述重命名对象包括更改第一命名空间内的对象名。
38. 如权利要求 36 所述的装置, 其特征在于, 所述重命名对象包括移动对象至第一命名空间内的另一位置。

## 用于与文件系统保持命名空间一致性的方法和系统

### 技术领域

本发明一般涉及计算机，尤其涉及文件系统。

### 背景技术

使用当代操作系统，诸如具有如 Windows® NTFS (Windows® NT 文件系统)、FAT、CDFS、SMB 重定向器文件系统的基础文件系统，或 WebDav 文件系统的 Microsoft 公司的 Windows® XP 操作系统，一个或多个文件系统过滤器驱动程序可被插入至接收用户 I/O 请求的 I/O 管理器和文件系统驱动程序之间。一般地，过滤器驱动程序（在此有时候简称为“过滤器”）是通过执行各种用户所需的关于文件的计算任务来增强基础文件系统的处理，这些任务包括诸如通过反病毒软件、文件系统配额提供程序、文件复制程序，以及加密/压缩产品发送文件系统 I/O（请求和数据）的任务。

例如，反病毒产品提供当文件复制产品执行文件系统级镜像时，查看至某个文件类型和来自某个文件类型（.exe、.doc 等）的 I/O 来查找病毒签名的过滤器。文件系统过滤器驱动程序的其它类型涉及系统还原（当要做出更改时备份系统文件以使用户可返回原始状态）、磁盘分配实施、开放文件备份、删除的文件的恢复、文件的加密等。如此，通过安装文件系统过滤器驱动程序，计算机用户可以用不改变实际操作系统或文件系统驱动程序代码而使组件的更新、替换、插入和移除成为可能的方式选择他们想要的和需要的文件系统特性。

文件系统过滤器可维护用于卷中的文件和目录的内部元数据。对于与过滤器关联的卷的修改可导致过滤器的内部元数据与卷的状态不同步。这个情况可导致过滤器行为错误或呈现为它不能执行想要的功能。

需要的是用于在由文件系统维护的选择的对象及与之关联的过滤器之间保持命名空间一致性的方法和系统。

### 发明概要

简单地说，本发明提供一种用于在由文件系统维护的选择的对象和与之关联的过滤器之间保持命名空间一致性的方法和系统。一处理请求访问文件系统的对象。过滤器监控选定类型的请求（或与之关联的操作）并确定该对象是否在与过滤器关

联的命名空间内。与过滤器关联的命名空间基于对象的更改而更新。

在本发明的一个方面中，请求包括重命名或删除操作。如果对象具有与之关联的策略或如果对象是这样一个对象的祖先，那么该对象在与过滤器关联的命名空间内。

在本发明的另一方面中，重命名操作可导致对象被移除和/或重命名。

在本发明的另一方面中，对象通过打开对象的句柄而被访问。句柄包括标识被用以访问对象的通信信道的信息。标记可被设置以指示当对象的所有句柄关闭时，对象应该被删除。如果在最后一个句柄关闭后标记保持设定，那么对象被删除。

在本发明的一方面中，保持命名空间一致性允许当对象被重命名时，过滤器应用对象上的策略并停止应用删除的对象上的策略。

当结合附图时，其它方面将从以下详细的描述中显现，在图中：

### **附图简述**

图 1 是框图，表现了可实现本发明的计算机系统；

图 2 是框图，表现了根据发明的各个方面，本发明可在其内操作的系统的组件的典型排列；

图 3 是框图，表现了根据发明的各个方面，本发明可在其内操作的系统的组件的另一典型排列；

图 4 是框图，表现了根据发明的各个方面，本发明可在其内操作的系统的组件的另一典型排列；

图 5 是框图，表现了根据本发明的各个方面，可被用以监控具有对其应用的策略的对象的数据结构；

图 6 是流程图，一般表现了根据本发明的各个方面，可发生以保持感兴趣的对象的过滤器命名空间和文件系统命名空间之间的一致性的动作；

图 7 是流程图，一般表现了根据本发明的各个方面，当重命名操作被接收时，对应于图 6 的框 630 可发生的动作；

图 8 是流程图，一般表现了根据本发明的各个方面，在更新数据结构时，对应于图 7 的框 745 可发生的动作；

图 9 是流程图，一般表现了根据本发明的各个方面，当对象被移动时，对应于图 8 的框 815 在添加节点时可发生的动作；

图 10 是流程图，一般表现了根据本发明的各个方面，当更新数据结构以移除

与旧对象名关联的前缀节点时，对应于图 8 的框 825 可发生的动作；

图 11 是流程图，一般表现了根据本发明的各个方面，对应于图 6 的框 640 在删除相关的操作期间可发生的保持过滤器命名空间和文件系统命名空间之间的一致性的动作；

图 12 是流程图，一般表现了根据本发明的各个方面，当对于在关闭操作后的删除的创建被接收时，对应于图 11 的框 1115 可发生的动作；

图 13 是流程图，一般表现了根据本发明的各个方面，当重命名操作被接收时，对应于图 6 的框 632 可发生的动作；

图 14 是流程图，一般表现了根据本发明的各个方面，当设置配置操作被接收时，对应于图 11 的框 1135 可发生的动作；及

图 15 是流程图，一般表现了根据本发明的各个方面，当清除操作被接收时，对应于图 11 的框 1145 可发生的动作。

## 发明的详细描述

### 典型操作环境

图 1 示出了一合适的计算系统环境 100 的示例，在该环境中可实现本发明。计算系统环境 100 只是一合适的计算环境的一例，而不是要提出对本发明使用或功能的范围进行限制。计算系统环境 100 也不应解释成对于在示例计算系统环境 100 中所示出的任一组件或其组合有任何依赖或需求。

本发明可用多个其它通用或专用计算系统环境或配置操作。可以适用于本发明使用的公知的计算系统、环境，和/或配置的实例包括，但不局限于，个人计算机、服务器计算机、手持或膝上型设备、多处理器系统、基于微处理器的系统、机顶盒、可编程消费者电器、网络 PC、小型机、大型计算机，及包括任何诸如以上系统或设备的分布式计算环境等。

本发明可以在诸如由计算机执行的程序模块的可实现计算机指令的一般上下文中被描述。一般地，程序模块包括完成特殊任务或实现特殊抽象数据类型的例行程序、程序、对象、组件、数据结构等。本发明也可以在分布式计算环境中实践，其中通过连接到一计算网络的远程处理设备来执行任务。在分布式计算环境中，程序模块可以定位于包括存储器存储设备的本地和远程计算机存储介质。

参见图 1，用于实现本发明的示例的系统包括一以计算机 110 形式出现的通用计算设备。计算机 110 的组件包括，但不局限于，处理单元 120、系统存储器 130，以及将包括系统存储器 130 在内的各种系统元件耦合至处理单元 120 的系统总线 121。系统总线 121 可以是多种总线体系结构类型中的任何一种，包括存储器总线或存储器控制器、外围总线和使用各种总线体系结构的任一种的局部总线。举例来说，而非限制，此类体系结构包括工业标准体系结构 (ISA) 总线、微通道体系结构 (MCA) 总线、增强型工业标准体系结构 (EISA) 总线、视频电子标准技术协会 (VESA) 局部总线，和也被称为 Mezzanine 总线的外围部件互连 (PCI) 总线。

计算机 110 通常包括各种计算机可读介质。计算机可读介质可以是任何计算机 110 能够访问的可用介质，包括易失性的和非易失性的介质、可移动的和不可移动的介质。举例来说，而非限制，计算机可读介质可以包含计算机存储介质和通信介质。计算机存储介质包括能以任何方法或技术实现的易失性的和非易失性的、可移动的和不可移动的介质，用于存储诸如计算机可读指令、数据结构、程序模块或其它数据等信息。计算机存储介质包括，但不局限于，RAM、ROM、EEPROM、闪存或其它存储技术，CD-ROM、数字化多功能光盘 (DVD) 或其它光盘存储、盒式磁带、磁带、磁盘存储器或其它磁存储设备，或任何其它可以被用来存储所需信息并且可以被计算机 110 访问的介质。通信介质通常体现为诸如载波或其它传送机制的已调数据信号中的计算机可读指令、数据结构、程序模块或其它数据，也包括任何信息传递介质。注意术语“已调制数据信号”是指在该信号中以编码信息的方式来设置或改变其一个或多个特征的信号。举例来说，而非限制，通信介质包括诸如有线网或直线连接的有线介质，和诸如声音、射频、红外线和其它无线介质的无线介质。任何以上所述的组合也可以包括在计算机可读介质的范围之内。

系统存储器 130 包括以诸如只读存储器 (ROM) 131 和随机存取存储器 (RAM) 132 的易失性和/或非易失性存储器的形式的计算机存储介质。包含如在启动期间帮助在计算机 110 内各元素之间传送信息的基本例行程序的基本输入输出系统 (BIOS) 133，通常存储在 ROM 131 中。RAM 132 通常包含可以被处理单元 120 立即访问和/或当前操作的数据和/或程序模块。举例来说，而非限制，图 1 示出了操作系统 134、应用程序 135、其它程序模块 136 和程序数据 137。



计算机 110 还可以包括其它可移动/不可移动、易失性/非易失性的计算机存储介质。仅作为示例，图 1 示出了读或写不可移动、非易失性磁性介质的硬盘驱动器 141、读或写可移动、非易失性磁性磁盘 152 的磁盘驱动器 151，以及读或写可移动、非易失性光盘 156 的光盘驱动器 155。其它可以使用在示例的操作环境中的可移动/不可移动、易失性/非易失性计算机存储介质包括，但不局限于，盒式磁带、闪存卡、数字多功能光盘、数字视频带、固态 RAM、固态 ROM 等。硬盘驱动器 141 通常通过诸如接口 140 的不可移动存储接口连接到系统总线 121，磁盘驱动器 151 和光盘驱动器 155 通常通过诸如接口 150 的可移动存储接口连接到系统总线 121。

以上讨论并且在图 1 中示出的驱动器及它们相关的计算存储介质为计算机 110 提供了计算机可读指令、数据结构、程序模块和其它数据的存储。在图 1 中，例如，示出硬盘驱动 141 存储操作系统 144、应用程序 145、其它程序模块 146，和程序数据 147。需要注意的是这些组件可以和操作系统 134、应用程序 135、其它程序模块 136 和程序数据 137 相同，也可以和它们不同。在此对操作系统 144、应用程序 145、其它程序模块 146 和程序数据 147 给出了不同的标号来说明至少它们是不同的拷贝。用户可以通过诸如键盘 162 和定位设备 161 这样的输入设备把命令和信息输入到计算机 110 中，定位设备 161 通常指如鼠标、跟踪球或触摸板。其它输入设备（图 1 中未示）可以包括麦克风、操纵杆、游戏垫、圆盘式卫星天线、扫描仪、手持 PC 或其他写字板的触摸屏等等。这些和其它输入设备通常由用户输入接口 160 连接到处理单元 120，上述输入接口 160 和系统总线 121 相耦合。但是上述和其它输入设备也可以由其它接口和总线结构连接到处理单元 120，诸如，并行端口、游戏端口或通用串行总线（USB）。监视器 191 或其它类型显示设备也可以通过诸如视频接口 190 的接口连接到系统总线 121。除监视器外，计算机 110 也包括其它外围输出设备，诸如可以通过一输出外围设备接口 195 等连接的扬声器 197 和打印机 196。

计算机 110 可以在网络化的环境中运行，使用逻辑连接连接到一个或多个诸如远程计算机 180 的远程计算机。远程计算机 180 可以是个人计算机、服务器、路由器、网络个人计算机、对等设备或其它共用网络节点，通常包括以上描述的和计算机 110 相关的多个或全部组件，尽管在图 1 中只示出了存储器储存设备 181。在图 1 中描绘的逻辑连接包括局域网（LAN）171 和广域网（WAN）

173, 但是也可以包括其它网络。这样的网络环境在办公室、企业范围的计算机网络、内联网和因特网中是普遍的。

当在 LAN 网络环境中使用时, 计算机 110 通过网络接口或适配器 170 连接到 LAN 171。当在 WAN 网络环境中使用时, 计算机 110 通常包括调制解调器 172 或通过诸如因特网的 WAN 173 建立通信的其他装置。调制解调器 172 可以是内置的或外置的, 可以通过用户输入接口 160 或其它适当的机制连接到系统总线 121。在一网络连接的环境中, 所述的和计算机 110 相关的程序模块或其中的一部分可以存储在远程存储器储存设备内。举例说明, 但非限制, 图 1 示出了驻留在存储设备 181 上的远程应用程序 185。可以理解的是所示的网络连接是示例的, 也可以使用在计算机间建立通信链路的其他装置。

### 保持命名空间一致性

图 2 是框图, 表现了根据发明的各个方面, 本发明可在其内操作的系统的组件的典型排列。这些组件包括一个或多个应用程序 205、应用编程接口 (API) 210、输入/输出 (I/O) 管理器 215、过滤器管理器 220、文件系统 225, 以及一个或多个过滤器 230-232。

应用程序 205 可通过 API210 向 I/O 管理器 215 做出文件系统请求 (例如, 通过函数/方法调用)。I/O 管理器 215 可确定哪个 I/O 请求应该被处理以实现每个请求以及发送每个 I/O 请求至过滤器管理器 220。I/O 管理器 210 也可把数据返回至应用程序 205 作为与文件系统请求继续、结束, 或中止的关联操作。

在一实现中, 当实例化的寄存器 (例如, 在他们初始化过程期间) 在过滤器管理器 220 内具有登记机制时候过滤器包括对象等。为了有效, 每个过滤器一般只登记文件系统对处理感兴趣的请求。为此, 作为登记的一部分, 每个过滤器通知过滤器管理器 220 其感兴趣的 I/O 请求的类型 (例如, 创建、读、写、关闭、重命名等)。例如, 一加密过滤器可登记读和写 I/O, 但是对于其它不需要加密或解密数据的 I/O 不进行登记。类似地, 配额过滤器只对对象创建和对象写感兴趣。

除了指定被感兴趣的 I/O 请求类型外, 过滤器可进一步指定过滤器是否应该在预回调时得到通知和发送对于每个 I/O 类型的回调。当与 I/O 请求关联的数据从 I/O 管理器 215 发送至文件系统 225 时, 调用预回调, 而当与 I/O 请求关联的数据从文件系统 225 发送至 I/O 管理器 225 时, 在 I/O 请求完成的期间, 调用后回调。

从每个 I/O 请求中,过滤器管理器 220 可以用合适于由过滤器 230-232 使用的统一格式创建数据结构。在下文中,该数据结构有时候称为回调数据。然后过滤器管理器 220 可调用和发送回调数据至每个已经登记接收对于由过滤器管理器 220 接收到的 I/O 类型回调的过滤器。任何登记以接收对于由过滤器管理器接收的 I/O 类型的回调的过滤器有时候称为登记的过滤器。

一般地,过滤器管理器 220 以登记的过滤器排序的次序顺序地发送与特定 I/O 请求类型关联的回调数据至每个登记的过滤器。例如,如果过滤器 230 和 232 被登记以接收所有读 I/O 请求的回调并且是以过滤器 230 在过滤器 232 之前处理这些请求的次序排列,那么在接收一个读 I/O 之后,过滤器管理器 220 可首先调用和发送回调数据至过滤器 230 并且在过滤器 230 处理完回调数据后,接着过滤器管理器 220 可调用和发送回调数据(如果在任何程度上被修改)至过滤器 232。

过滤器可附加在一个或多个卷上。也就是说,过滤器可被登记已被调用和接收用于与仅仅一个或一个以上卷相关的 I/O 的回调数据。

过滤器可产生以后可被发送至其它过滤器的它自己的 I/O 请求。例如,一反病毒过滤器希望在文件被打开之前读一文件。过滤器可停止来自其它地方发送的 I/O 请求然后可指示过滤器管理器报告 I/O 请求的状态码(例如,成功或失败)。过滤器可在存储器内存储数据并且在磁盘上保留该数据。一般地,过滤器可被创建以执行任何可由内核模式或用户模式处理执行的以及可以是反应性的(例如,直到接收到 I/O 请求才行动)和/或原发性的(例如,初始化它自己的 I/O 请求或异步地执行其它具有由 I/O 管理器 215 处理的请求的动作)动作组。

在一实施例中,过滤器可以如图 3 示出的堆栈的方式排列,图 3 是框图,表现了根据发明的各个方面,本发明可在其内操作的系统的组件的另一典型排列。在该实施例中,过滤器 305-307 中的每个过滤器都可处理 I/O 请求(修改的或未被修改的)和发送请求至另一过滤器或其它栈内的组件。例如,响应于从应用程序 205 中的一个接收到的读请求,I/O 管理器 215 可提交 I/O 请求并发送该请求至过滤器 305。过滤器 305 可检测 I/O 请求并且确定过滤器 305 不对 I/O 请求感兴趣,然后发送未更改的 I/O 请求至过滤器 306。过滤器 306 可确定过滤器 306 将基于 I/O 请求执行某些动作并且然后发送 I/O 请求(更改的或未被更改的)至过滤器 307。过滤器 307 可确定过滤器 307 不对 I/O 请求感兴趣并且发送请求至文件系统 235。

在文件系统 235 处理 I/O 请求后,它发送结果至过滤器 307。一般地,结果以

I/O 请求被处理相反的次序发送(例如, 首先发送至过滤器 307, 然后至过滤器 306, 然后至过滤器 305)。过滤器 305-307 中的每个过滤器检测结果, 确定过滤器是否对结果感兴趣, 并且可在发送结果(更改的或未被更改的)至另一过滤器或组件之前基于结果执行动作。

在本发明的另一实施例中, 过滤器可以如图 4 中示出的堆栈/管理的方式排列, 图 4 是框图, 表现了根据发明的各个方面, 本发明可在其内操作的系统的组件的另一典型排列。在该配置中, 某些过滤器与过滤器管理器关联而其它过滤器则不与过滤器管理器关联。过滤器管理器 220 与其它过滤器(例如, 过滤器 305 和 307)放置在一个栈内。

可以容易意识到的是在不背离本发明精神和范围的情况下, 过滤器可以由许多其它配置所实现。在某些实施例中, 过滤器包括任何检测应用程序与文件系统之间的 I/O 的对象, 并且这样的对象可更改、完成, 或中止 I/O 或执行其它基于 I/O 的动作。

返回图 2, 文件系统 235 可包括一个或多个卷, 该卷可本地或远程地定位在一个或多个机器上, 应用程序 205 在所述机器上执行。

某些过滤器监控在某些目录内的文件。例如, 配额过滤器可实施在特定目录上的配额。如另一例子, 数据筛选过滤器可拒绝在某些目录内某些文件类型(例如, 由后缀名如 .MP3, .JPG 等可取得版权的文件)的创建。这些过滤器可包括识别涉及的目录的命名空间。

某些文件系统请求(例如, 重命名或删除)可更改由过滤器监控的对象的文件系统命名空间。例如, 配额过滤器可被配置以实施在名为 C:\DIR\QUOTA 的对象上一千兆字节的配额。对象名称和应用至对象上的配额可被存储在非易失性存储器上保留的元数据文件中。应用程序可把对象 C:\DIR 重命名为 C:\DIR2 或可从 C:\DIR\QUOTA 移动至 C:\QUOTA。为了继续在 QUOTA 对象上实施配额, 配额过滤器监控重命名并在每次重命名影响配额过滤器实施配额的对象时, 更新配额过滤器的元数据文件。

在某些操作系统中, 重命名操作可重命名对象和/或移动对象。如此, 通过监控重命名操作, 过滤器可捕获名字的重命名、对象的移动, 或两者。在其中重命名文件和移动文件是独立操作的操作系统内, 过滤器需要监控这两种操作以保持命名空间与文件系统的命名空间一致性。

此外，应用程序可删除对象，例如 C:\DIR 或 C:\DIR\QUOTA。为了避免监控已经被删除的对象，当现存对象被删除，并且具有相同名称的新的对象在它的位置上被重新创建时，不实施策略，以及为了减少在监控这样的对象上使用的元数据的大小，过滤器可监控删除并且相应地更新它的元数据。

图 5 是框图，表现了根据本发明的各个方面，可被用以监控具有对其应用的策略的对象的数据结构。如在此涉及的，监控的对象包括目录、文件、任何可存储在文件系统上的事物等。

此数据结构包括每个感兴趣的对象的节点并且有时称为前缀树。表现的典型对象包括根目录“\”、\DIR、\DIR1\QUOTA1、\DIR1\QUOTA1\QUOTA2、\DIR2、\DIR2\DIR3，以及 \DIR2\DIR3\QUOTA3。对象 \DIR1\QUOTA1、\DIR1\QUOTA1\QUOTA2，以及 \DIR2\DIR3\QUOTA3 有各自应用到它们的策略 P1、P2，和 P3。策略可包括在磁盘空间上由对象及它的后代（例如，子目录）消耗的配额、对象及它的后代中所允许的文件，针对对象及它的后代所允许和/或不被允许的内容，等等。当感兴趣的对象在文件系统上被重命名或删除时，图 5 所示的数据结构需要被更新。这样的更新可涉及移除节点、添加额外的节点，以及更改以下更详述的节点内容。

一般地，如果重命名或删除对象将影响在数据结构内寻找对象或策略的应用，那么对象是所感兴趣的。例如，如果 \DIR2 和它的子目录被删除，那么策略 P3 不再需要被应用至 \DIR2\DIR3\QUOTA3。此外，节点 DIR2、DIR3，以及 QUOTA3 将不再被需要应用策略 P3 并且可以被删除以保存空间和减少对于任何剩余策略的查找时间。

如另一例子，如果 \DIR1\QUOTA1\QUOTA2 在文件系统上被重命名为 \QUOTA5，这可影响在数据结构内寻找新对象。由于在过滤器应用策略至特定操作之前，过滤器需要直到操作是否可影响任何已经为之建立策略的对象，所以这是重要的。为了在数据结构内寻找新的对象（例如，以使过滤器确定它是否需要在特定操作上实施策略），新的节点可被创建并且与新对象关联，然后与旧节点 QUOTA2 关联的策略（例如，P2）可被复制到新节点，然后旧节点 QUOTA2 可被删除。接着策略 P2 对于新对象被实施。响应于对象的更改而添加和删除节点在以下结合附图 8-10 将做详细描述。

应用（例如，服务或其它）可把具有与对象关联的策略的对象发送至过滤器。应用可提供这样的对象列表或根据需要添加和/或删除策略。在接收该列表或指令

后，过滤器可创建新的数据结构或修改现存的数据结构以说明所感兴趣的对象。

数据结构可在启动时或使用该数据结构的过滤器实例化时被重构。这可使用以上描述的保留在非易失性存储器（例如，磁盘）上的元文件来完成。

数据结构可在一个以上的过滤器间共享。例如，两个或两个以上的过滤器可对被重命名或删除的对象实施策略。两个或两个以上的过滤器可共享数据结构，而不是对每个过滤器维护独立的数据结构备份。一个或一个以上的过滤器可被指定以保持数据结构与文件系统同步。

此外，即使过滤器可共享数据结构，但是它们不希望监控所有相同的对象。为此，额外的数据可被维护以指示每个过滤器所感兴趣的节点。

图 6 是流程图，一般表现了根据本发明的各个方面，可发生以保持感兴趣的对象的过滤器命名空间和文件系统命名空间之间的一致性的动作。在框 605 处，过程开始。在框 610，过滤器接收所感兴趣的 I/O 操作。在管理的过滤器的情况下，过滤器可已经预先登记以接收已接收到的类型的 I/O 操作。在堆栈的过滤器的情况下，过滤器可确定 I/O 是可影响文件系统命名空间的。在一实现中，某些可影响文件系统命名空间的 I/O 操作包括重命名和删除。

在框 615，做出 I/O 操作是否涉及所感兴趣的对象的确定。如果是，处理转至框 625；否则处理转至框 620，在此过程返回。参考图 5，所感兴趣的对象是由任何数据结构的节点所表现的对象。在接收与所感兴趣的 I/O 操作关联的数据后，过滤器在数据结构上执行查找以确定 I/O 操作是否涉及其中任何节点。如果不是（例如，查找失败），过滤器返回。

在框 625，做出 I/O 操作是否是重命名操作的确定。如果是，处理转至框 630；否则处理转至框 635。在另一实施例中，与框 615 和 610 关联的动作被颠倒。也就是说，首先做出 I/O 操作是否涉及所感兴趣的对象的确定，然后做出 I/O 操作是否是所感兴趣的确定。在该实施例中，如果 I/O 不涉及所感兴趣的对象或如果 I/O 不是所感兴趣的 I/O，过程返回。

在框 630，过滤器命名空间被适当地更新为如结合图 7-10 所详细描述的那样。在框 632，如果与对象关联的流上下文如结合附图 13 所详细描述的那样存在，则该流上下文被更新。流上下文包括数据，从中可存储或检索有关对象的信息。流上下文可为对象所创建，例如，当对象被标记为删除时。即使对象可被标记为删除，在它被删除之前可别重命名。为了跟踪重命名中对象的名称，对象的新名称存储在

流上下文中。每次对象被重命名，流上下文被更新以包括对象的新名称。然后，当对象被完全删除时，过滤器在流上下文中查找所删除的对象最近的名称并且从过滤器的命名空间内删除适当的一个节点或多个节点。

在框 635，做出操作是否是删除相关的操作的确定。如果是，处理转至框 640；否则，处理转至框 645。删除相关的操作可包括在关闭操作、一组配置操作，或如结合图 11 详细描述的清操作上删除的创建。在框 640，过滤器命名空间被适当更新为如结合图 10 所详细描述的那样。

在框 645，任何其它过滤器动作可被执行。例如，配额过滤器可通过禁止或允许 I/O 请求继续以实施配额。在某些实现中，与框 645 关联的动作在框 610 之后发生。这可允许过滤器禁止操作和停止任何对象的更改。在框 620，过程返回。结合图 6 描述的过程在每次所感兴趣的 I/O 被接收时重复。

图 7 是流程图，一般表现了根据本发明的各个方面，当重命名操作被接收时，对应于图 6 的框 630 可发生的动作。在框 705，过程开始。在框 710，做出重命名操作是在预回调还是在后回调阶段的确定。如果操作是在预回调阶段（也就是说，在 I/O 请求发送至文件系统之前），处理转至框 715；否则，如操作在后回调阶段，处理转至框 730。实质上，在更新数据结构之前，过程等待直到操作成功。

在框 715，回调上下文被分配并且与操作关联。回调上下文可包括任何数据并提供发送有关操作的数据至后回调阶段的方式。当后回调阶段对特定操作发生时，与操作关联的回调上下文可被访问。

在框 720，在操作被执行之前的对象名（也就是说，旧名称）被存储在回调上下文中。在框 725，后操作回调被强制调用，实质上表明过滤器指示它希望在文件系统上的操作被执行完之后以及 I/O 结果返回应用之前被调用。

在框 730 的后操作阶段中，做出操作是否失败的确定。如果是，处理转至框 750；否则处理转至框 735。如果操作失败，不需要更新数据结构。

在框 735，从回调上下文中检索对象的旧名称。该旧名称被用以在维护过滤器命名空间的数据结构中寻找节点。在框 740，询问文件系统，查找对象的新名称，该新名称可包括新对象名称和/或对象路径。在框 745，数据结构被更新，如结合图 8 详细描述的那样。在框 750，释放回调上下文。在框 755，过程返回。

图 8 是流程图，一般表现了根据本发明的各个方面，在更新数据结构时，对应于图 7 的框 745 可发生的动作。在框 805，过程开始。在框 810，做出对象是否被

移动的确。如果是，处理转至框 815；否则，处理转至框 830。如果对象被移动，旧节点可被删除并且新节点被创建以如由框 815、820，以及 825 所指示的更新过滤器命名空间和策略。如果仅仅对象名被更改而不是它的路径被更改，如在节点内指示的对象名可被更改为由框 830 所指示的新对象名。

在框 835，被移动的节点的后代节点可被适当更新。在一实现中，完整的路径和对象名存储在每个节点中。例如，参考图 5，QUOTA2 节点可存储 \DIR1\QUOTA1\QUOTA2。这个可为了提高执行速度或其它原因而进行。当对象名和/或节点路径被更改时，是该节点后代节点的每个节点可被更新以包括包括新对象名和/或与更改的节点关联的路径的新前缀路径。在其它实现中（例如，完整的路径和对象名没有在节点内存储并且不需要在后代节点内被更新），框 835 可被跳过。

在框 840，过程返回。

图 9 是流程图，一般表现了根据本发明的各个方面，当对象被移动时，对应于图 8 的框 815 在添加节点时可发生的动作。在框 905，过程开始。在框 910，选择对象新路径的第一个成分。如在路径中被使用，成分涉及相应于对象的祖先对象的路径的一部分并且包括根目录。例如，\DIR1\QUOTA1\QUOTA2 的成分是根（也就是说，“\”）、DIR1、QUOTA，以及 QUOTA2。该第一成分是根。

在框 915，做出是否存在对应于选择的组件的节点的确。如果是，处理转至框 925，新节点不需要被创建；否则，处理转至框 920。在框 920，一相对于组件的新节点被创建。

在框 925，做出是否当前选择的成分是新路径的最后一个成分的确。如果是，处理转至框 905，在此处理返回。否则，处理转至框 930，在此下一成分被选择并且与框 915-925 关联的动作被重复直到最后一个成分被处理。可以意识到的是结合图 9 所述的过程实质上修改数据结构，以包括在没有存储于数据结构中的新路径中用于任何成分的节点。

图 10 是流程图，一般表现了根据本发明的各个方面，当更新数据结构以移除与旧对象名关联的前缀节点时，对应于图 8 的框 825 可发生的动作。一般地，当节点被移动到数据结构内的新位置时，所有不再必要的节点从数据结构中移除。如果一节点没有与其后代节点关联的策略并且该节点本身不具有与之关联的策略，那么该节点不再是必要的。

过程开始于框 1005。在框 1010，与旧对象关联的节点被选择。在框 1015，旧



节点的策略被移除，因为它们先前已经被复制到新节点中。在框 1020，做出当前选择的节点是否具有与之关联的策略的确定。如果是，处理转至框 1045；否则，处理转至框 1025。

在框 1025，做出节点是否具有子节点的确定。如果是，处理转至框 1045；否则，处理转至框 1030。由于节点被维护用于具有策略以及它们祖先的对象，否则则不被维护，则如果节点具有任何子节点，这就表明节点的后代节点中至少一个与策略关联。例如，如果不具有策略的节点有两个子节点并且其中一个子节点被重命名以使它不再是节点的后代节点，除非第二个子节点也被移除，该节点将不被删除。

在框 1030，如不再需要它，则删除节点。在框 1035，做出节点是否有父节点的确定。如果是，处理转至框 1040，在此选择父节点。从框 1040，处理继续到框 1020 并且与框 1020-1035 关联的动作被重复。可以意识到的是，实际上，这些动作递归地沿数据结构向上删除节点直到它们找到本身与策略关联或是具有与策略关联的子节点的节点。

在框 1045，过程返回。

可以意识到的是图 6-10 的流程图表现一种当对文件系统发生重命名时，命名空间数据结构可被更新以一致的方法，并且该方法在不背离本发明的精神或范围的情况下，重命名由多种方式实现期间，保持命名空间数据结构与文件系统的命名空间一致。

图 11 是流程图，一般表现了根据本发明的各个方面，对应于图 6 的框 640 在删除相关的操作期间可发生的保持过滤器命名空间和文件系统命名空间之间的一致性的动作。在框 1105，过程开始。

在框 1110，做出操作是否是在关闭操作后对于删除的创建。如果是，处理转至框 1115；否则，处理转至框 1130。创建操作可创建对象或打开已经存在的对象。对象可为关闭后删除而被创建。当对象为关闭后删除而创建时，表示该对象在被关闭后应该被删除。也就是说，对象可以被创建、存在，以及继续存在直到所有具有对象的句柄的进程关闭它们的句柄。在所有句柄被关闭后，对象被删除。在框 1115，用于创建的动作被执行，如结合图 12 所详细描述。

在框 1130，做出操作是否是一组配置操作的确定。如果是，处理转至框 1135；否则，处理转至框 1140。在一实现中，对象可具有与之关联的指示是否该对象应该在关闭所有它的句柄后被删除的标记。该标记可被设定（也就是说，指示该对象

应该删除)或清除(也就是说,指示该对象不应该被删除)。此外,标记可由每个具有对象句柄的进程所设定或清除。如此,一进程可设定标记仅仅使标记由另一进程所清除。对象保留或被删除取决于当具有对象的句柄的最后一个进程关闭句柄时标记的配置(例如,被设定或被清除)。在框 1135,涉及设定对象配置的动作被执行,如结合图 14 所详细描述。

在框 1140,做出操作是否是清除操作的确定。如果是,处理转至框 1145;否则,处理转至框 1155。每次处理关闭对象的句柄时,清除操作可被提交。然而,如先前提到的,对象实际上不被删除直到具有对象的句柄的最后一个进程关闭句柄。即使在最后一个进程已经关闭对象后,如果例如,在最后关闭时,用户删除对象的标记指示该对象不应该在关闭后被删除,则该对象可不被删除。在框 1145,执行清除动作,如结合图 15 所详细描述。

在框 1155,过程返回。结合图 11 所述的过程可在每次过滤器接收对所感兴趣的对象的删除相关 I/O 操作时被重复。

图 12 是流程图,一般表现了根据本发明的各个方面,当对于在关闭操作后的删除的创建被接收时,对应于图 11 的框 1115 可发生的动作。在框 1205,过程开始。

在框 1210,做出操作是否是在预操作阶段的确定。如果是,处理转至框 1215;否则,处理转至框 1220。

在框 1215,强制执行后操作回调以使过滤器在操作的后操作阶段接收回调。在框 1220,做出操作是否失败的确定。如果是,处理转至框 1240;否则,处理转至框 1225。

在框 1225,流上下文被分配(如果它没有存在)并且与对象关联。在框 1230,对象名存储在流上下文中。在框 1235,删除标记在流上下文中被更新。对于用于关闭后删除而被创建的对象,标记可指示当对象的所有句柄关闭时,对象被删除。此类标记可不被后面的进程所修改。当对象的所有句柄关闭时,用于关闭后删除所被打开的对象被删除。

对于不为关闭后删除而打开的对象,标记可被设定或清除(通过一组配置操作)以指示对象应该或不应该在关闭后被删除。然后,标记可由后面的设置配置操作更改。在所有句柄被关闭后,对象是否被删除取决于当最后一个对象的句柄关闭时,标记的状态。

在框 1240，过程返回。

图 13 是流程图，一般表现了根据本发明的各个方面，当重命名操作被接收时，对应于图 6 的框 632 可发生的动作。在框 1305，过程开始。

在框 1310，做出操作是否在预操作阶段的确定。如果是，处理转至框 1330；否则，处理转至框 1315。

在框 1330，强制执行后操作回调以使过滤器在操作的后操作阶段接收回调。在框 1315，做出操作是否失败的确定。如果是，处理转至框 1335；否则，处理转至框 1320。

在框 1320，做出与对象关联的流上下文是否存在的确定。如果是，处理转至框 1325；否则，处理转至框 1335。如果例如，对象不为关闭后删除而打开并且在指示对象是否应该在关闭后被删除的标记上没有配置被设置或清除，流上下文可不存在。

在框 1325，名称在流上下文中被更新以反映被更改的对象的名称。这是有用的，例如，在稍后确定名称更改是否影响任何节点上。

在框 1335，过程返回。

图 14 是流程图，一般表现了根据本发明的各个方面，当设置配置操作被接收时，对应于图 11 的框 1135 可发生的动作。在框 1405，过程开始。

在框 1410，做出操作是否在预操作阶段的确定。如果是，处理转至框 1415；否则，处理转至框 1420。

在框 1415，强制执行后操作回调以使过滤器在操作的后操作阶段可接收回调。在框 1420，做出操作是否失败的确定。如果是，处理转至框 1440；否则，处理转至框 1425。

在框 1425，做出与对象关联的流上下文是否存在的确定。如果是，处理转至框 1435；否则，处理转至框 1430，在此创建流上下文并且在其中存储对象名。流上下文是与对象关联的。

在框 1435，删除标记被更新以反映对象的删除配置。如以上描述的，进程可设定标记以指示对象应该在关闭后被删除或清除标记以指示在关闭后控制器不应该被删除。在框 1440，过程返回。

图 15 是流程图，一般表现了根据本发明的各个方面，当清除操作被接收时，对应于图 11 的框 1145 可发生的动作。在框 1505，过程开始。

在框 1510，做出操作是否在预操作阶段的确定，如果是，处理转至框 1515；否则，处理转至框 1520。

在框 1515，强制执行后操作回调以使过滤器在操作的后操作阶段接收回调。在框 1520，做出操作是否失败的确定。如果是，处理转至框 1550；否则，处理转至框 1525。

在框 1525，做出与对象关联的流上下文是否存在的确定。如果是，处理转至框 1530；否则，处理转至框 1550。在框 1530，做出对象是否是过滤器感兴趣的对象的确定。如果是，处理转至框 1535；否则，处理转至框 1550。

在框 1535，做出删除标记是否被设定为删除或对象是否为关闭后删除而打开的确定。如果这些条件的任意一条被满足，处理转至框 1540；否则，处理转至框 1550。在一实现中，当对象为关闭后删除而打开时，不会在稍后指示对象不在关闭后被删除。

在框 1540，做出对象是否被删除的确定。由于多个进程可打开一个对象以及对象只在所有进程已经关闭对象的句柄后被删除，即使所有其它条件已经被满足，对象也可不被删除。在本发明的一个实施例中，当对象已经被删除时，操作系统可通知过滤器（而不是过滤器自己检测）。在这样的实施例中，结合图 15 所述的方法可包括确定对象是否是所感兴趣的并且合当地删除节点。

在框 1545，节点和任何相关的节点被适当地删除。如果节点是其它在结合图 5 所述的数据结构内的节点的祖先，其它节点也被删除。此外，类似于那些结合图 10 所述的动作可被执行，以删除节点的祖先节点以及不再需要的节点。在框 1550，过程返回。

可以意识到的是在不背离本发明精神和范围的情况下，结合图 11-15 所述的动作可被其它动作替代，并且根据操作系统操作的方式以不同的次序执行。例如，某些操作系统可不具有标记以指示文件是否应该在所有对于文件的句柄关闭后被删除。而是，这样的操作系统可提供独立的删除操作以删除对象。在这样的操作系统中，用于检测删除的更改的方法可包括当对象被删除时的监控以及删除与对象关联的节点（如果有的话）。

图 12-14 的后操作调用的强制执行可在后操作调用默认发生的实现中被跳过。

可以从以上详细的描述中看出，本发明提供了一种用于保持命名空间与文件系统的一致性的方法和系统。尽管本发明允许各种修改和可供选择的构造，某些示例

---

性的实施例在此结合附图被展示并且详细的描述。然而，可以理解的是，本发明不受公开的特定形式的限制，而是相反，本发明涵盖所有在本发明的精神和范围内的修改、可供选择的构造，和同等物。

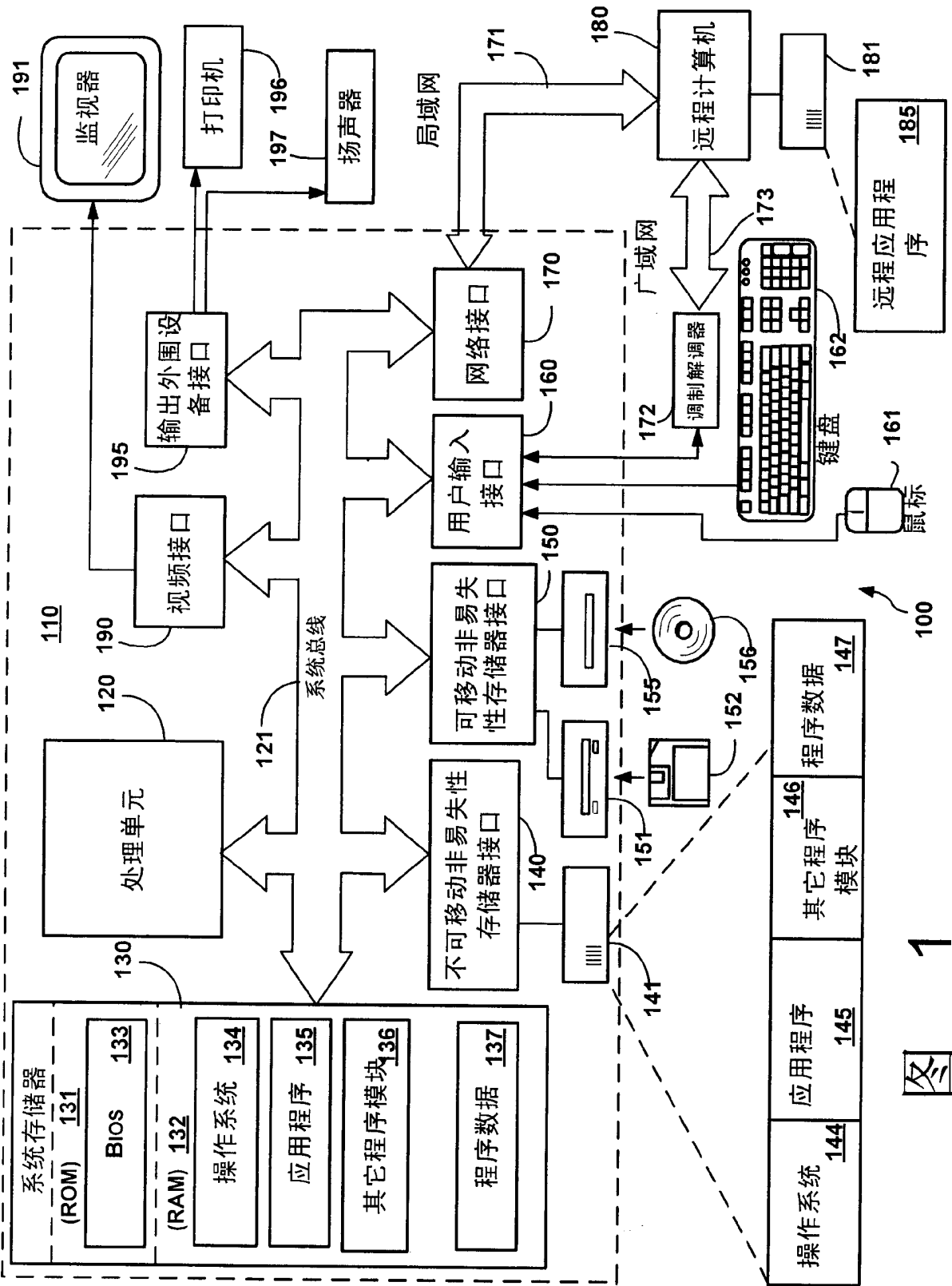


图 1

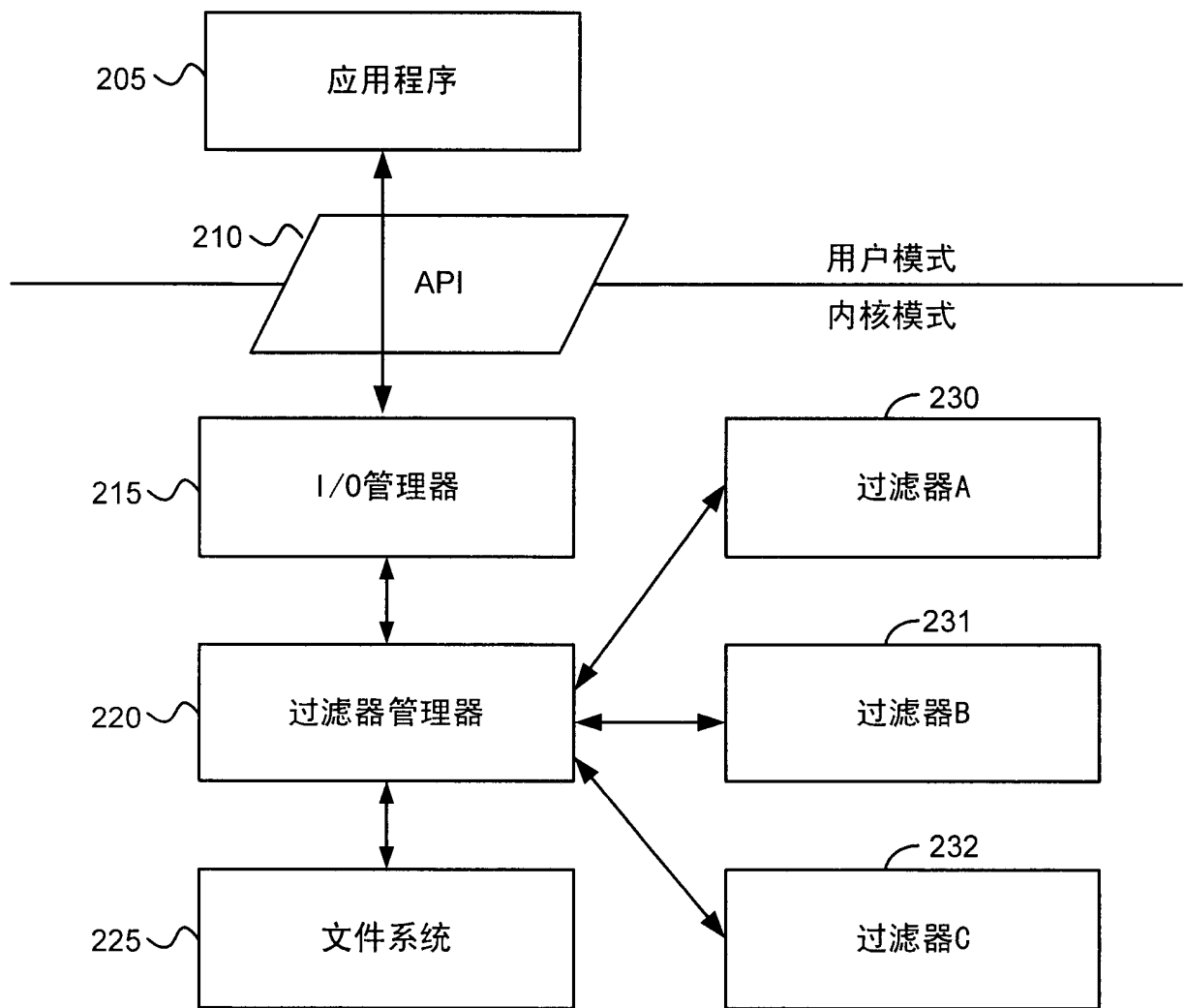


图 2

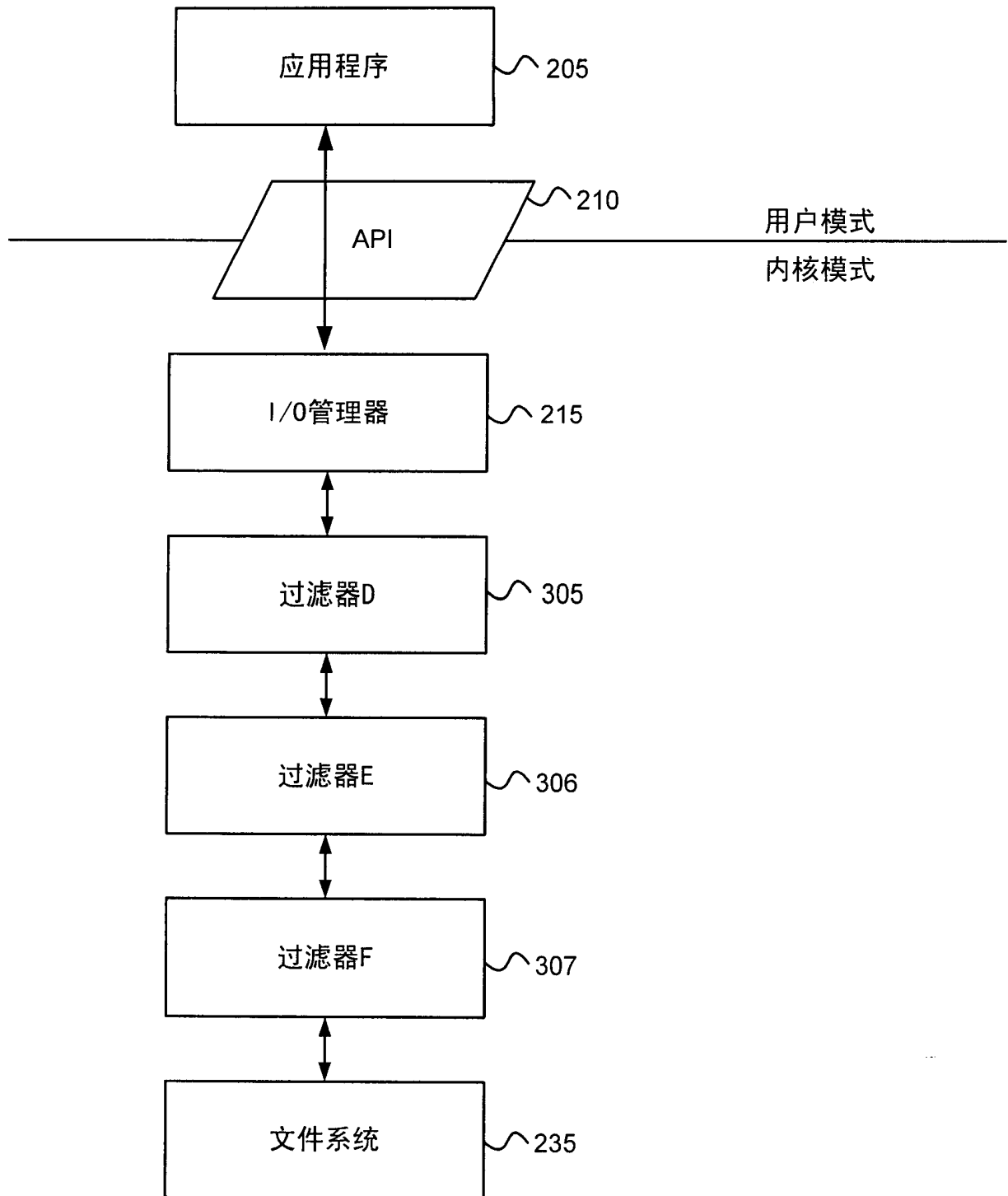


图 3



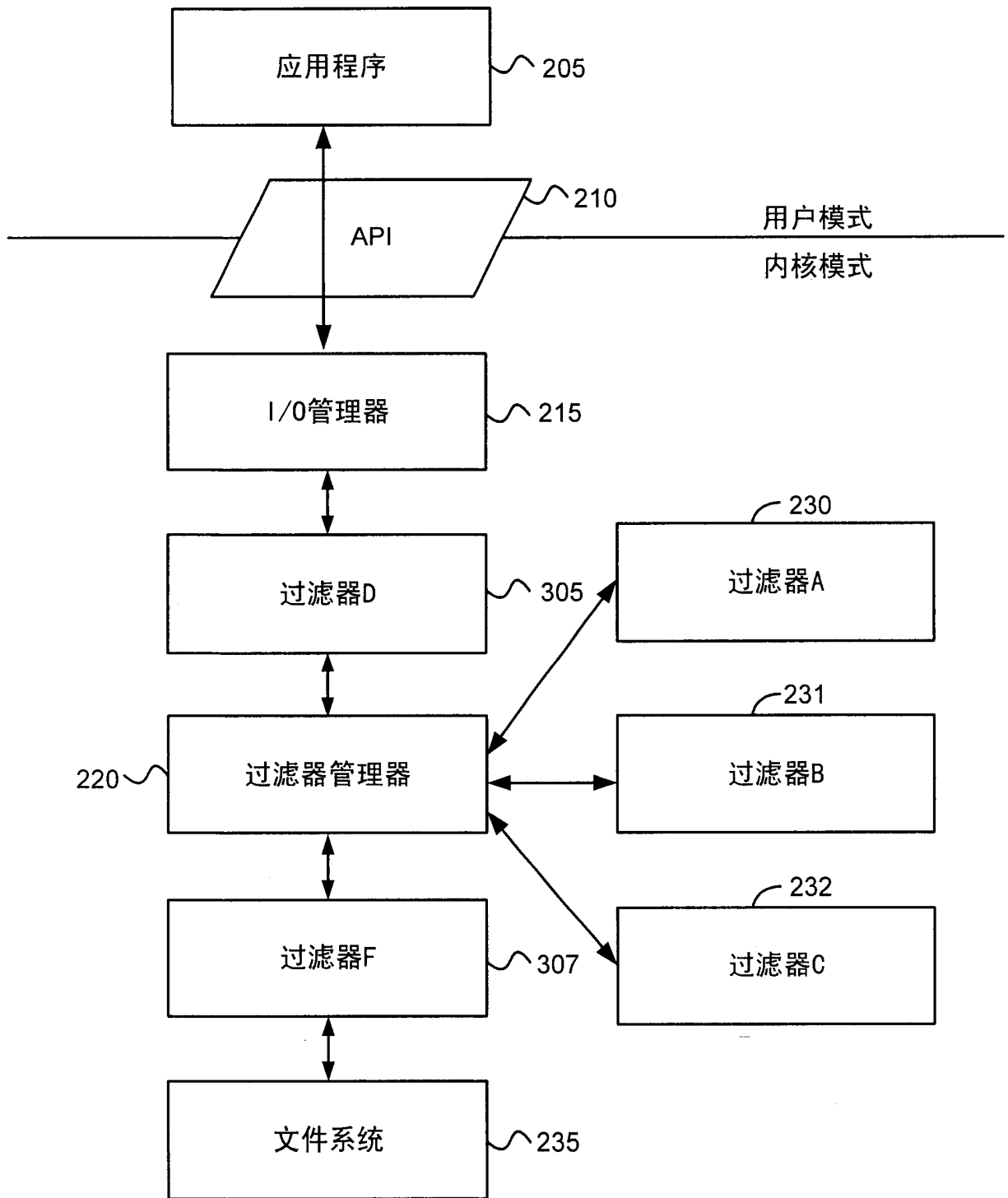


图 4

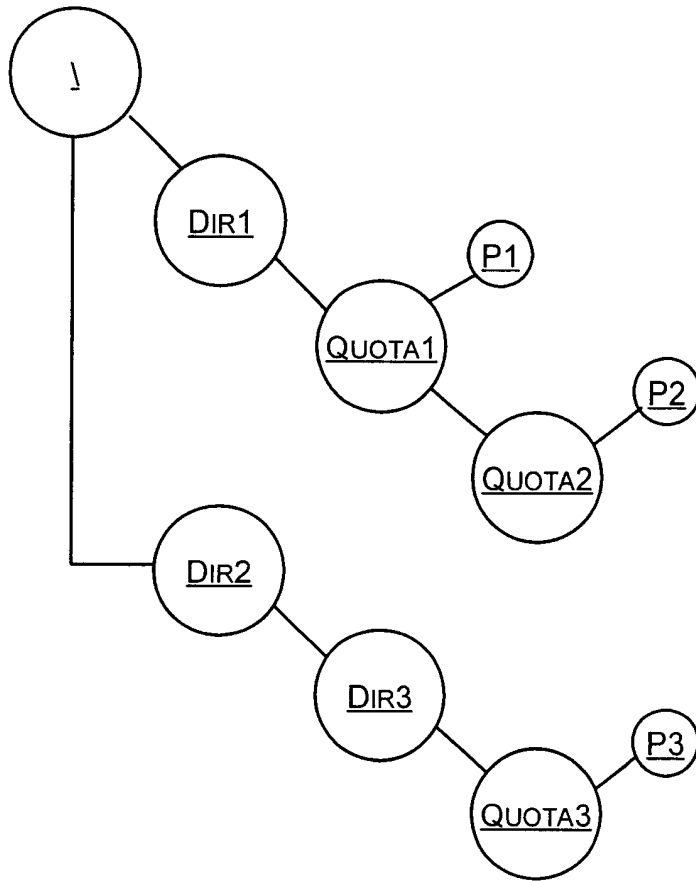


图 5

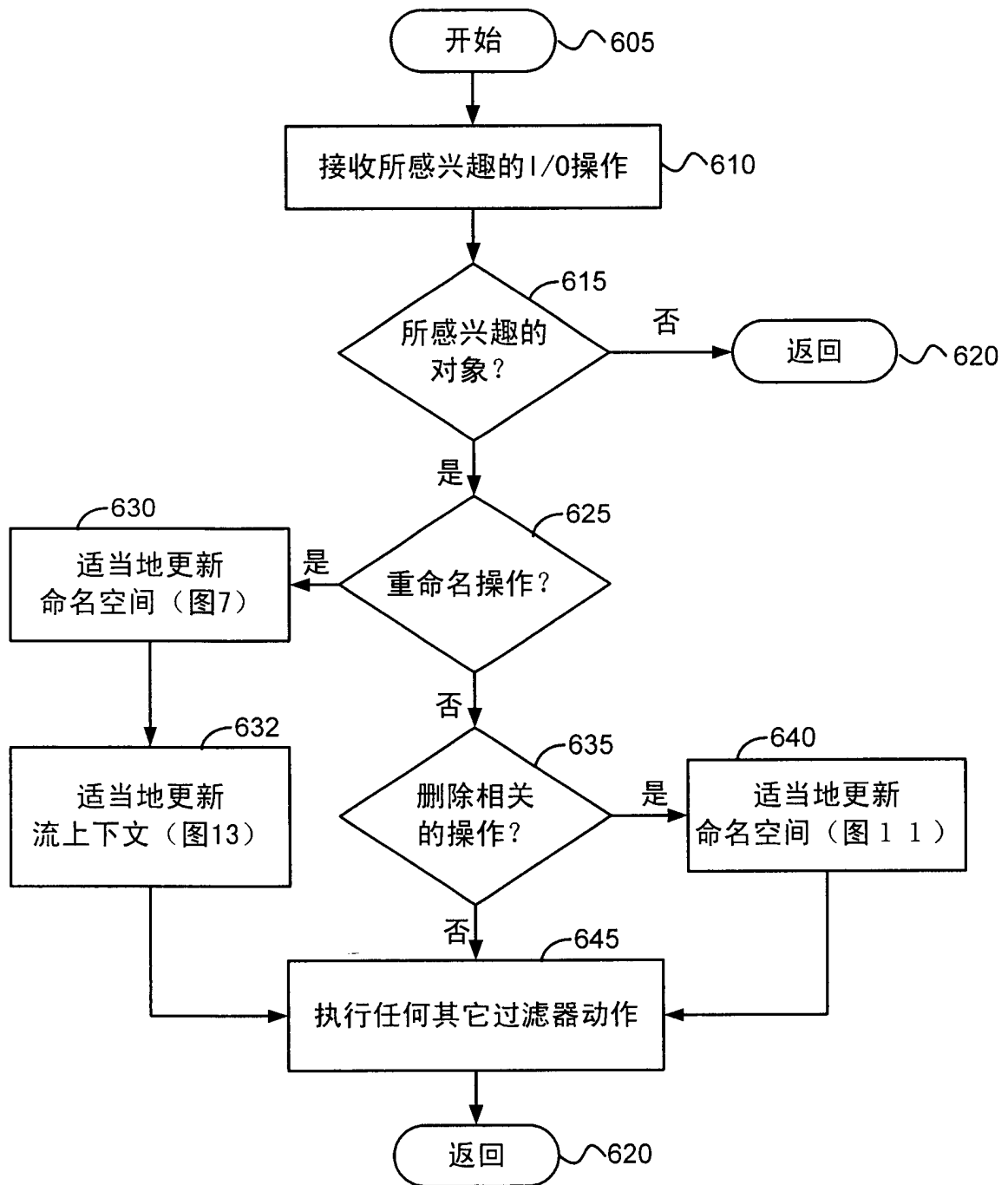


图 6

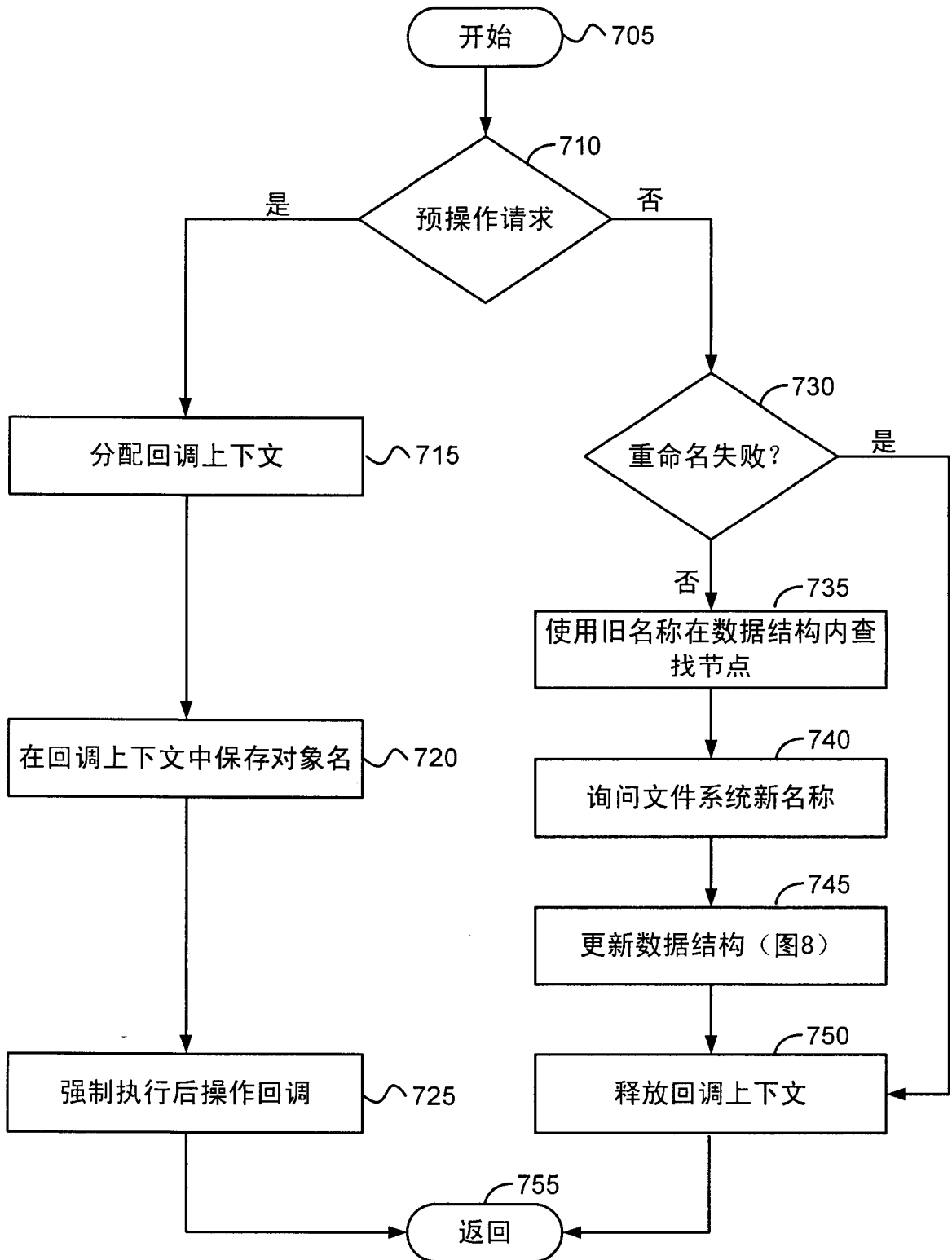


图 7

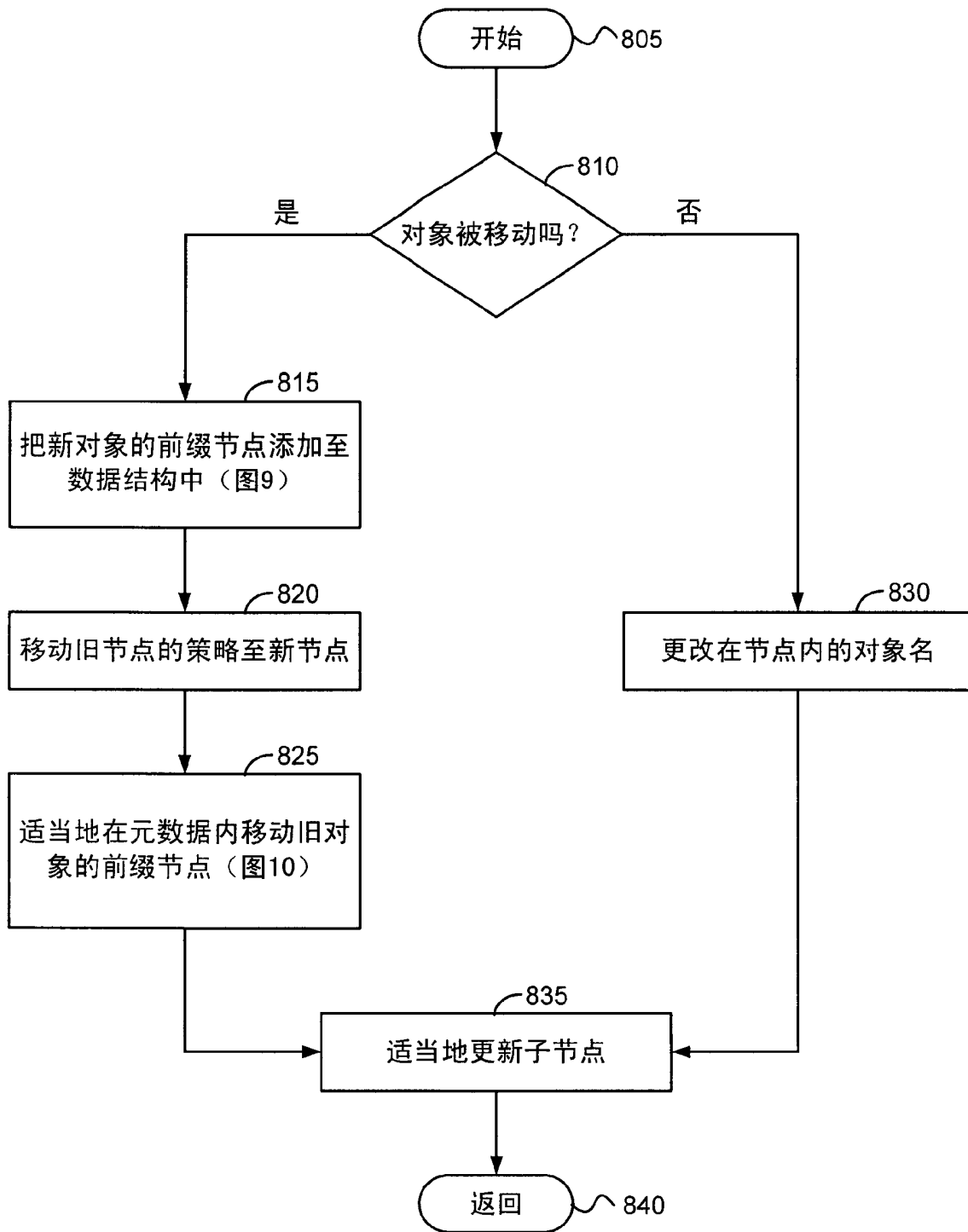


图 8

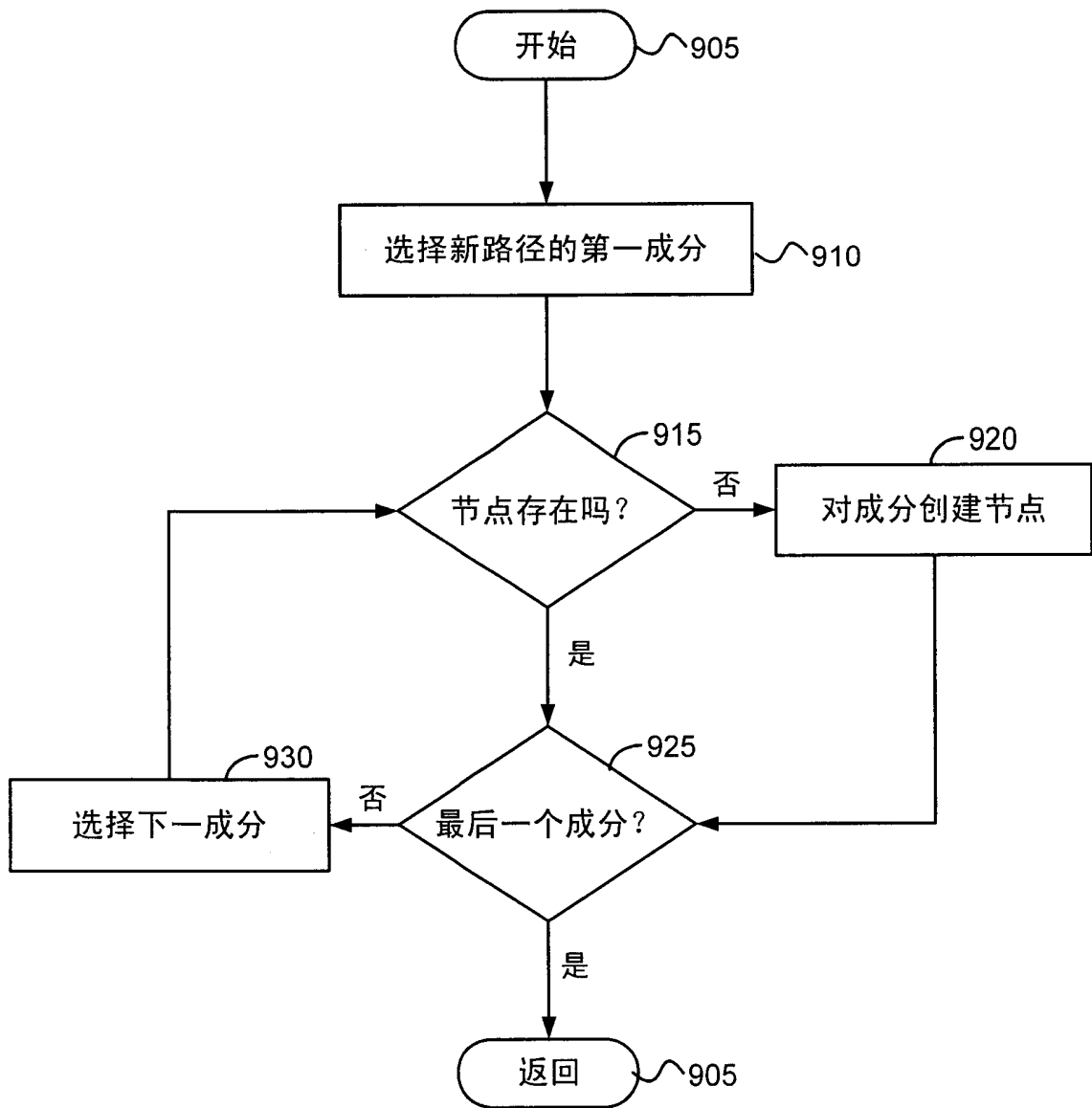


图 9

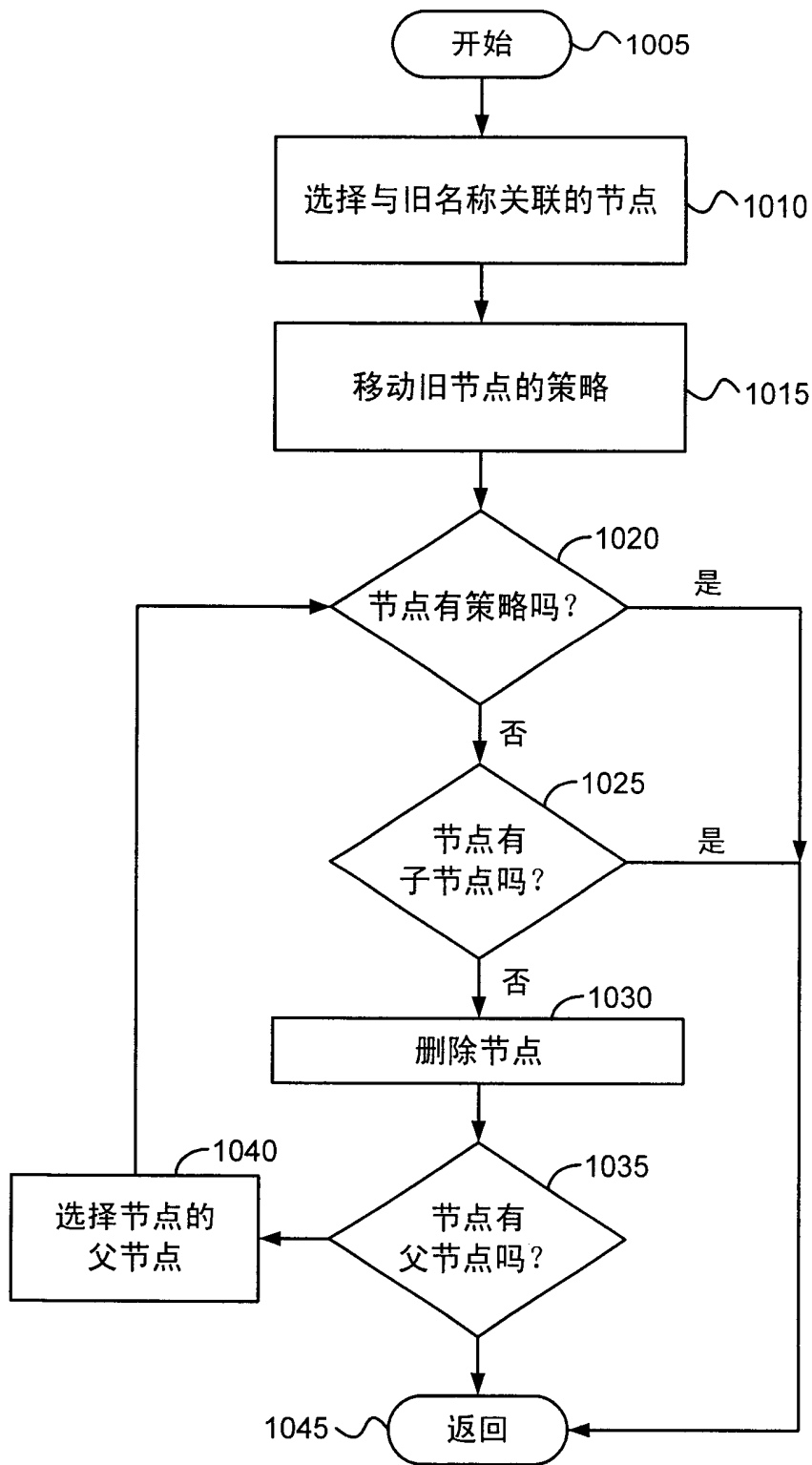


图 10

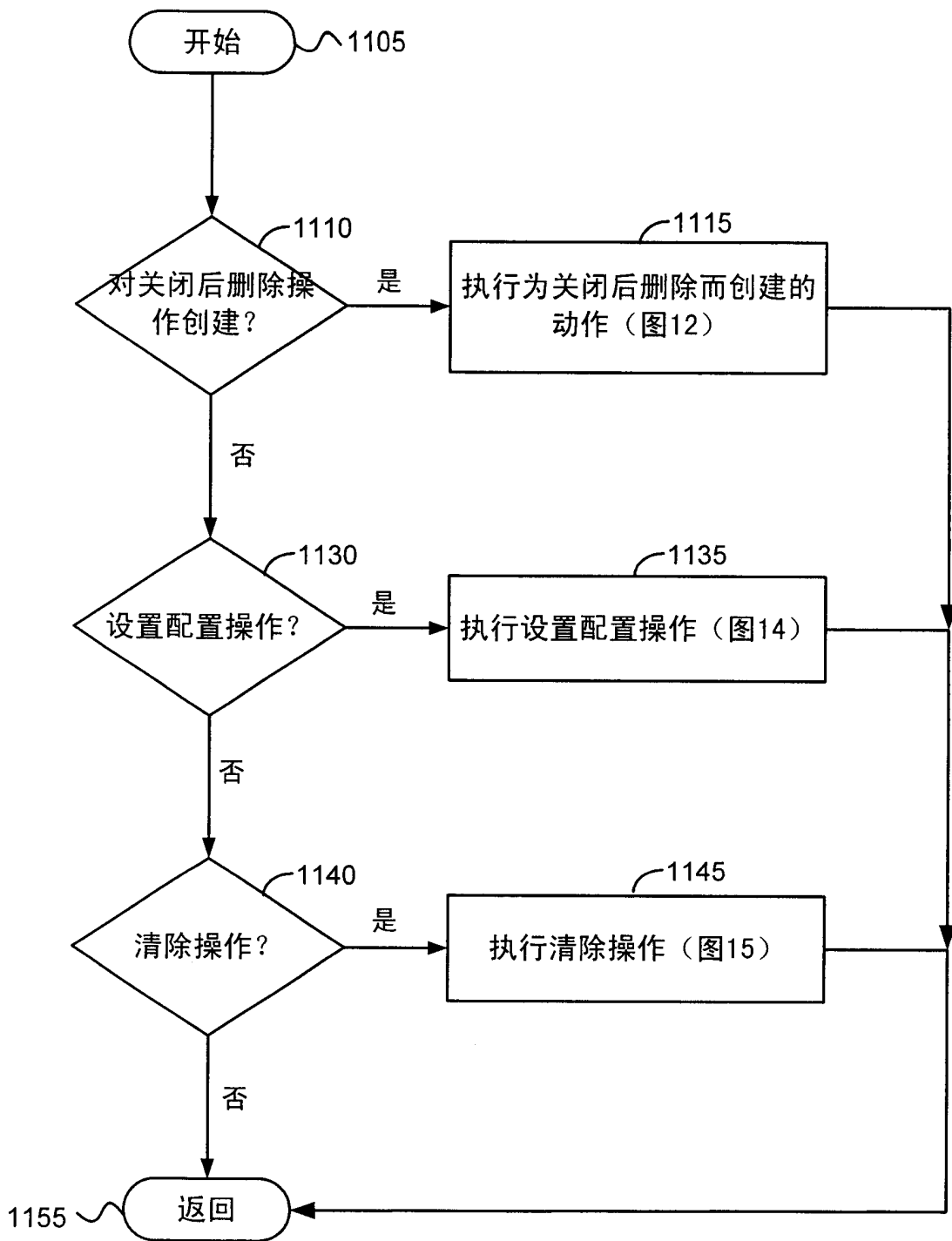


图 11



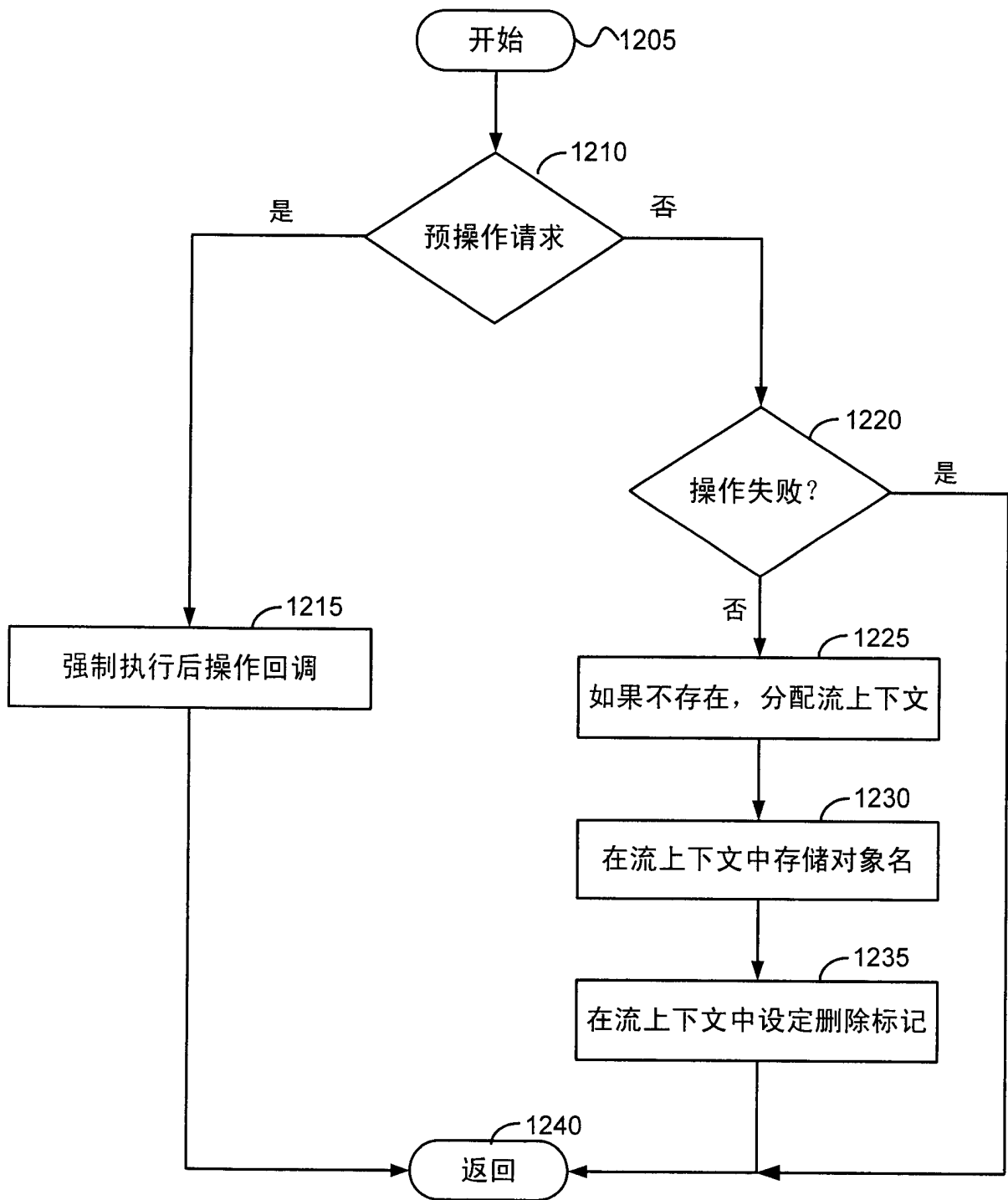


图 12

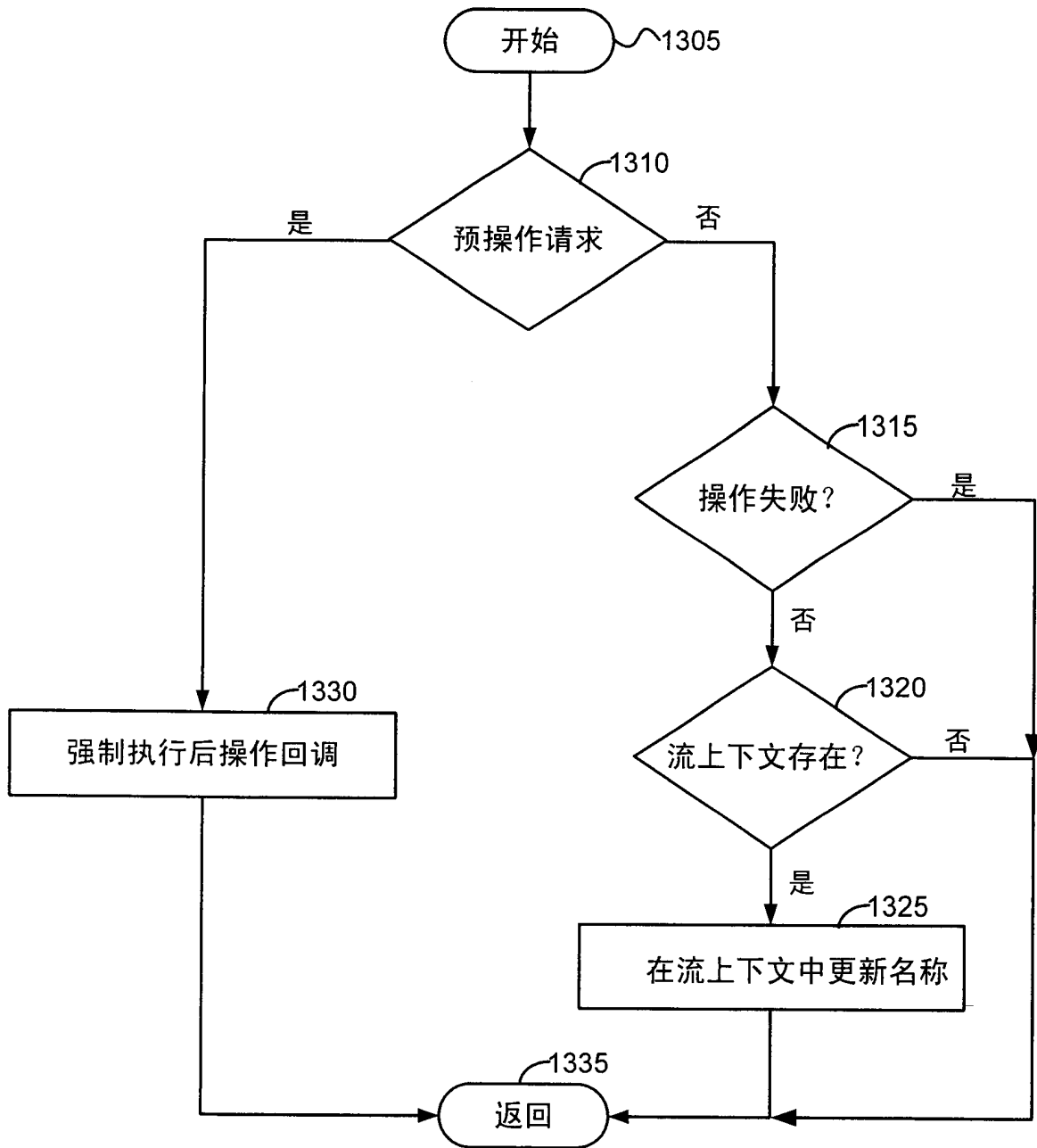


图 13

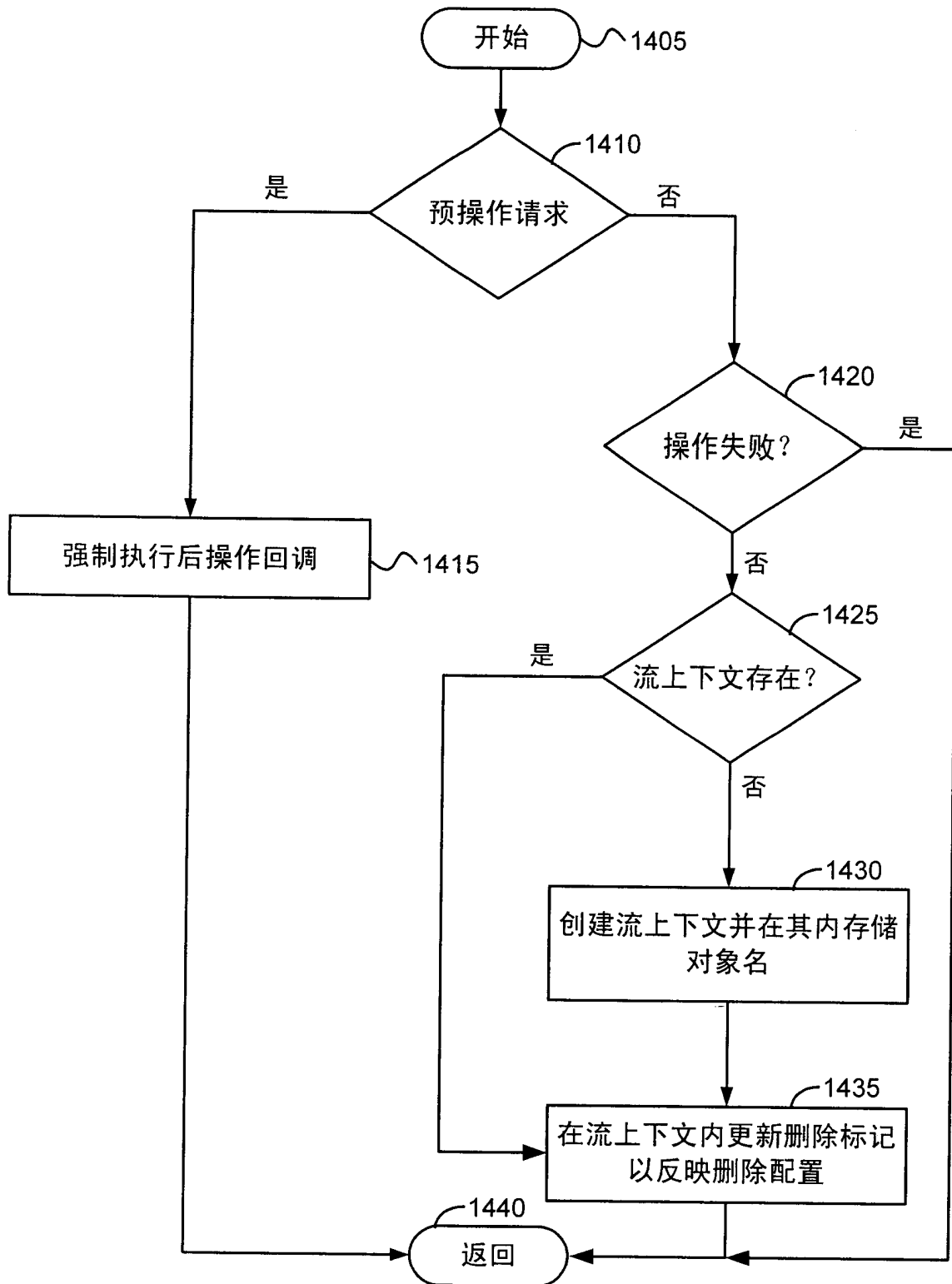


图 14

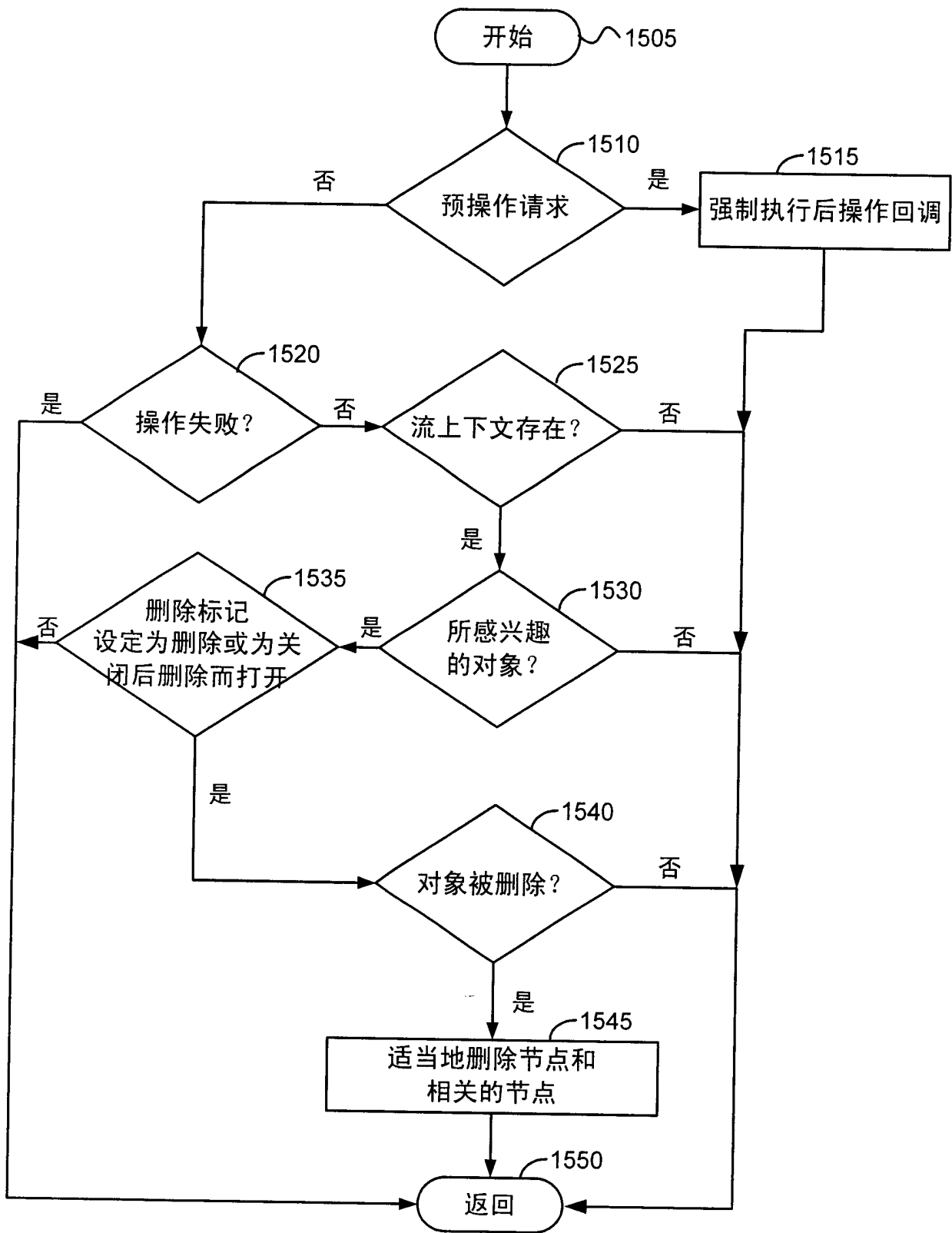


图 15