



(12) 发明专利

(10) 授权公告号 CN 1773509 B

(45) 授权公告日 2010.12.22

(21) 申请号 200510124811.3

(22) 申请日 2005.11.14

(30) 优先权数据

10/985,907 2004.11.12 US

(73) 专利权人 微软公司

地址 美国华盛顿州

(72) 发明人 A·G·拜比 A·K·诺里

B·S·拉曼 T·P·麦基

W·R·史密斯

(74) 专利代理机构 上海专利商标事务所有限公

司 31100

代理人 张政权

(51) Int. Cl.

G06F 17/30 (2006.01)

(56) 对比文件

CN 1517906 A, 2004.08.04, 全文.

US 6615224 B1, 2003.09.02, 说明书摘要、说

明书第5栏第54行至第6栏第56行、说明书附图

1.

全文.

全文.

US 5544360 A, 1996.08.06, 说明书摘要、说
明书第4栏第58行至第6栏第60行、说明书附图

2.

审查员 程小梅

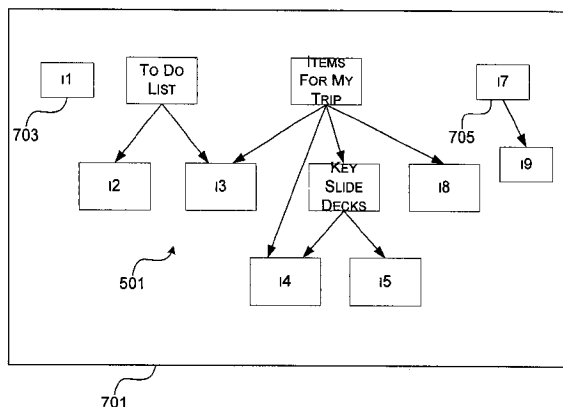
权利要求书 2 页 说明书 10 页 附图 8 页

(54) 发明名称

计算机文件系统

(57) 摘要

本发明描述了不将项的生存周期与该项在文件
系统组织结构中的内容合并的文件系统。另外，
该文件系统的组织结构不限于目录树，而是可以
使用任何有向无环图 (DAG)。一旦存储到文件系
统中，项将被维护，直到由文件系统的客户机肯定
地删除，而不论该项是否是 DAG 的一部分。项可
以被放置在文件区域中，文件区域是客户机用来
控制项生存周期以及将项组织进用户选择的任何
DAG 结构的概念上的工作空间。项可以并发地存
储在多个 DAG 中，且每一文件区域可以含有一个
或多个独立的 DAG。项在 DAG 中的位置也可以用于
管理诸如名字空间、安全性、保密性和读 / 写属
性等文件属性。



1. 一种文件系统管理器,所述文件系统管理器使用项位置数据和项组织数据来管理存储在文件系统系统中的项,所述文件系统管理器包括:

用于从所述文件系统中删除第一指定的项的装置,所述装置包括管理所述文件系统管理器的删除操作,所述删除操作从所述文件系统中删除第一指定的项;以及

用于从所述文件系统的组织结构中移除第二指定的项的装置,所述装置包括管理所述文件系统管理器的移除操作,所述移除操作从所述文件系统的组织结构中移除第二指定的项,而不从所述文件系统中删除第二指定的项;

用于查询第二数据表中的浮动或顶层的项,以查找在组织结构内不存在的任何项的装置;

其中,项位置数据是存储在第一数据表中的,且项组织数据是存储在第二数据表中的,且第二数据表中存储了存储在所述文件系统系统中的项之间的层次关系;

其中,第二数据表中的项是按照多个包含一个或多个树的有向无环图排列,且每一有向无环图包含未被第二数据表中的其它项的持有链接作为目标的浮动或顶层的项。

2. 如权利要求 1 所述的文件系统管理器,其特征在于,所述第一数据表是与所述第二数据表分开的。

3. 如权利要求 1 所述的文件系统管理器,其特征在于,所述用于从所述文件系统中删除第一指定的项的装置包括从所述第一数据表和所述第二数据表中删除对所述第一指定的项的任何引用。

4. 如权利要求 3 所述的文件系统管理器,其特征在于,所述用于从所述文件系统中删除第一指定的项的装置还包括在删除之前基于所述第二数据表中关于所述第一指定的项的条目向所述第二数据表添加新的条目。

5. 如权利要求 1 所述的文件系统管理器,其特征在于,所述用于从所述文件系统的组织结构中移除第二指定的项的装置包括从所述第二数据表中删除对所述第二指定的项的任何引用,而不从所述第一数据表中删除对所述第二指定的项的任何引用。

6. 如权利要求 1 所述的文件系统管理器,其特征在于,所述用于从所述文件系统中删除第一指定的项的装置是可以通过第一应用程序编程接口来调用的。

7. 如权利要求 6 所述的文件系统管理器,其特征在于,所述用于从所述文件系统的组织结构中移除第二指定的项的装置是可以通过不同于所述第一应用程序编程接口的第二应用程序编程接口来调用的。

8. 如权利要求 1 所述的文件系统管理器,其特征在于,还包括基于所述项组织数据的安全数据。

9. 如权利要求 1 所述的文件系统管理器,其特征在于,所述文件系统管理器基于所述项组织数据为每一项构造名字空间。

10. 如权利要求 1 所述的文件系统管理器,其特征在于,所述文件系统包括:
第一数据表,它存储所述文件系统内的至少第一、第二和第三项的项位置数据;以及
第二数据表,它存储所述第一和第二项的项组织数据,但是不包括所述第三项的项组织数据。

11. 如权利要求 10 所述的文件系统管理器,其特征在于,所述第一数据表和所述第二数据表是分开的表。

12. 如权利要求 10 所述的文件系统管理器,其特征在于,所述第一数据表为每一项存储唯一的项 ID、项名和项的存储位置。

13. 如权利要求 10 所述的文件系统管理器,其特征在于,所述第二数据表中的每一条目定义了两个存储的项之间的关系。

14. 如权利要求 13 所述的文件系统管理器,其特征在于,所述关系包括父-子关系。

15. 如权利要求 14 所述的文件系统管理器,其特征在于,所述第二数据表的所有条目可以被共同表示为有向无环图。

16. 如权利要求 13 所述的文件系统管理器,其特征在于,所述第二数据表存储将所述第一项标识为所述第二项的父项的第一条目。

17. 如权利要求 16 所述的文件系统管理器,其特征在于,所述第二项基于所述第一项和所述第二项之间的关系从所述第一项继承安全许可。

18. 如权利要求 1 所述的文件系统管理器,其特征在于,每一项的生存周期相对于存储在所述文件系统中的其它项是独立于每一项的组织的。

19. 如权利要求 1 所述的文件系统管理器,其特征在于,所述文件系统确定每一项独立于该项与存储在所述文件系统中的其它项的关系的生存周期。

20. 如权利要求 1 所述的文件系统管理器,其特征在于,所述文件系统存储第一项,所述第一项对存储在所述文件系统中的其它项没有父-子关系。

21. 一种用于管理存储在文件系统中的项的方法,所述方法包括以下步骤:

利用管理操作接收从所述文件系统中删除第一项的第一请求,所述管理操作管理文件系统管理器的删除操作,所述删除操作从所述文件系统删除第一项;

响应于第一请求,从与所述文件系统相关联的位置信息以及从与所述文件系统相关联的组织信息中删除对所述第一项的引用;

利用管理操作接收从所述文件系统中移除第二项的第二请求,所述管理操作管理文件系统管理器的移除操作,所述移除操作从所述文件系统的组织结构中移除第二项;

响应于第二请求,从与所述文件系统相关联的组织信息中删除对所述第二项的引用,并且保持与所述文件系统相关联的位置信息中对所述第二项的引用不变;

响应于第三请求,查询未被其它项的持有链接作为目标的浮动或顶层的项,以查找在组织结构内不存在的任何项,

其中,所述位置信息是存储在所述第一数据表中的,且所述组织信息是存储在第二数据表中的,且第二数据表中存储了存储在所述文件系统中的项之间的层次关系,其中第二数据表中的项是按照多个包含一个或多个树的有向无环图排列,且每一有向无环图包含所述浮动或顶层的项。

计算机文件系统

[0001] 本专利文档公开内容的一部分包括服从版权保护的材料。版权所有人不反对任何人按照其在(美国)专利商标局的专利文件或记录中的形式对专利文档或专利公开内容进行复制,但版权所有人保留其它所有权利。

技术领域

[0002] 本发明一般涉及计算机系统的文件系统。更具体地,本发明提供一种文件系统,该文件系统不将存储在该文件系统中的项的生存周期与由该文件系统所使用的底层组织结构合并,从而允许存储在该文件系统中的项的每一个在该组织结构中具有零个、一个或多个父项而没有被删除的风险。

背景技术

[0003] 计算机在诸如磁盘驱动器等存储设备上存储文件。磁盘驱动器仅提供类似于空文件箱等存储数据的场所。正如空文件箱不伴随文件的任何预定义的文件编排系统(即,文件箱的用户必须例如通过按照字母顺序排列文件来自己创建文件编排系统或组织结构),硬盘驱动器默认也仅是空的存储空间。仅靠其自身,访问硬盘驱动器上的数据的唯一的方法是,或者通过指定数据的物理位置(例如,通过指定存储文件的硬盘驱动器的柱面、磁头以及扇区),或者通过其在磁盘上的逻辑位置(例如,第 21,671 块)。一旦硬盘驱动器被安装在计算机上,计算机使用文件系统以可容易访问的方式来跟踪存储在硬盘驱动器上的文件。

[0004] 已知的文件系统过度地限制文件系统的操作系统和用户如何能够组织文件系统文件。即,已知的文件系统一般要求用户在文件和目录的树中组织项,其中目录实际上是由文件系统识别的特殊类型的文件。即使文件系统支持另外的数据结构,这种能力一般也不展示给文件系统的客户(即,终端用户和应用程序)。图 2 示出了现有文件系统的典型组织结构 201 的简要的示例。如在图 2 中所示,已知的文件系统使用树来组织目录(使用圆形端示出)和文件(使用方形端示出)。在树结构中,项的位置和组织是合并的;每一项必须在一个且仅一个目录中。这就是用户必须如何组织他或她的文件,且用户不能够在没有创建项的新副本的情况下将一个项置于多个组织中。

[0005] 典型的文件系统使用两个彼此结合的表或数据库来组织文件。第一张表或数据库是识别文件存储在诸如硬盘驱动器等存储设备上的物理位置的查找表。第二张表定义了文件的组织结构。这些表在此一般分别被称为位置表(LOC)和组织表(ORG)。组织表存储关于持有链接,即一个项是另一个项的父项的信息。某些文件系统可将位置表和组织表组合成单张表或结构,但仍旧要求存在两张表的元素以使文件系统能够正确地运作。例如,在由微软公司销售的 NT 文件系统,即 NTFS 中,主文件表既用作位置表又用作组织表。类似地,Unix 文件系统,即 UFS,使用一种 i 个节点的表,该表既用作文件的位置表又用作组织表。目录作为特殊类型的文件存储,其中目录“文件”在目录内部存储文件名列表和其各自的 i 个节点。

[0006] 在这些和其它已知的文件系统中,只要文件位于如由组织表中的持有链接定义的至少一个位置上,即,存在指向该文件的至少一个持有链接,文件系统就保持将该文件存储在物理存储设备上。即,如果从组织表中删除对文件的持有链接,且不再有指向该文件的持有链接,那么文件系统移除位置表中该文件的条目(不考虑该文件在存储设备上是否被物理地重写)。存储设备然后可以使用该存储空间来写入新的数据。例如,如果用户想要“删除”图 2 所示的文件 C:\PROGRAMS\MICROSOFT\OFFICE\WORD\FILE.DOC,那么文件系统首先从组织表中移除该文件的条目。如果该文件在位置表中没有其它条目,即该文件没有存储在其它地方,那么文件系统移除位置表中该文件的条目,由此为其它数据释放了空间。通常,文件系统维护任何给定项具有的祖先(被称为“持有链接”)的数量的引用计数。当移除项上的最后一个持有链接时,通过移除位置表中该文件的引用来“删除”该项。然而,从终端用户观点来看,这是不合需要的,因为它限制了文件系统可以使用的组织结构。

[0007] 由于上述约束和限制(例如,作为将文件从树中移除的副产物的删除),文件系统不允许客户机以不同于目录和文件的树形层次的数据结构来组织数据。用户想要能够以各种组织数据结构来对项进行组织和去除组织,而不涉及删除给定的项。如果项的生存周期与其在文件系统中的组织相分离,这将是本领域中的进步。即,提供不将项的生存周期与组织结构合并的文件系统将是本领域中的进步,在该文件系统中,对项进行组织和去除组织的动作不影响其生存周期。这样,提供这样一种文件系统,它不限制操作系统和/或用户能够用于组织文件的数据结构的类型,也不会仅仅因为文件从文件系统内的所有组织结构中移除或者没有指向该文件的持有链接而删除文件,将是本领域中的进步。

发明内容

[0008] 下文提供了本发明的简化的概述,以提供对本发明的某些方面的基本理解。该概述不是本发明的广泛的概观。它不旨在标识本发明的关键或重要的元素,也不描绘本发明的范围。下文的概述仅以简化的形式提供本发明的某些概念,以作为下文提供的更详细的描述的序言。

[0009] 为了克服上述的现有技术中的一个或多个限制,和/或克服在阅读和理解本说明书之后显而易见的其它限制,本发明一般针对一种实施为存储在计算机可读介质上的数据项和计算机可执行指令的文件系统。数据项一般指的是可以存储在文件系统中的任何数据,包括但不限于,文件、文件夹、数据、音乐等。该文件系统使用存储至少用于存储在文件系统内的第一、第二和第三数据项的项位置信息的第一数据表,并使用存储第一和第二数据项的组织信息但不存储第三数据项的组织信息的第二数据表。

[0010] 该文件系统的文件系统管理器软件模块可以被实施为存储在计算机可读介质上的计算机可执行指令。该文件系统管理器基于项位置数据和项组织数据来管理存储在文件系统中的数据项。该文件系统管理器使用用于从文件系统中删除项的第一子例程,并使用用于从文件系统中移除项的第二子例程。

[0011] 该文件系统管理器执行一种用于管理存储在文件系统中的项的方法,该方法包括接收从文件系统中删除第一项的第一请求,并响应于第一请求,从与文件系统相关联的位置信息以及从与文件系统相关联的组织信息中删除对第一项的引用。该文件系统管理器接收从文件系统中移除第二项的第二请求,并响应于第二请求,从与文件系统相关联的组织

信息中删除对第二项的引用,但是不从与文件系统相关联的位置信息中删除对第二项的引用。

[0012] 本发明的另一方面提供一种用于存储数据项的文件系统,其中该文件系统被实施为存储在计算机可读介质上的计算机可执行指令。在该文件系统中,数据项生存周期独立于该文件系统中数据项的任何概念组织内的数据项的位置。

附图说明

[0013] 考虑附图,通过参考下文的描述,可以获得对本发明及其优点的更完整的理解,附图中,相同的参考标号指示相同的特征,附图中:

[0014] 图 1 示出了根据本发明的说明性实施例适用于实现媒体用户界面的通用操作环境。

[0015] 图 2 示出了已知文件系统的典型组织结构。

[0016] 图 3 示出了根据本发明的说明性实施例的位置表 (“LOC”)。

[0017] 图 4 示出了根据本发明的说明性实施例的组织表 (“ORG”)。

[0018] 图 5 示出了根据图 4 所示的组织表的有向无环图 (DAG) 组织结构。

[0019] 图 6 示出了根据本发明的说明性实施例的文件系统的框图。

[0020] 图 7 示出了根据本发明的说明性实施例的文件区域。

[0021] 图 8 示出了定义图 7 的组织结构的组织表。

[0022] 图 9 示出了根据本发明的说明性实施例的文件区域。

[0023] 图 10 示出了定义图 9 的组织结构的组织表。

[0024] 图 11 示出了根据本发明的说明性实施例的组织表。

[0025] 图 12 示出了根据图 11 所示的组织表的文件区域。

[0026] 图 13 示出了根据本发明的说明性实施例用于管理存储在文件系统中的项的方法。

具体实施方式

[0027] 在下文对于各个实施例的描述中,参考附图,附图形成了本发明的一部分,并作为说明示出可在其中实现本发明的各种实施例。可以理解,可以使用其它实施例,且可以进行结构和功能的修改而不背离本发明的范围。

[0028] 示例性计算环境

[0029] 图 1 示出了可在其上实现本发明的合适的计算系统环境 100 的示例。计算系统环境 100 只是合适的计算环境的一个示例,并不旨在对本发明的使用范围或功能提出任何限制。也不应该把计算环境 100 解释为对示例性操作环境 100 中示出的任一组件或其组合有任何依赖性 or 要求。

[0030] 本发明可用众多其它通用或专用计算系统环境或配置来操作。适合在本发明中使用的公知的计算系统、环境和 / 或配置的示例包括,但不限于,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、可编程消费者电子产品、网络 PC、小型机、大型机、包含上述系统或设备中的任一个的分布式计算机环境等。

[0031] 本发明可在诸如由计算机执行的程序模块等的计算机可执行指令的通用语境下

描述。一般而言,程序模块包括例程、程序、对象、组件、数据结构等,它们执行特定任务或实现特定抽象数据类型。本发明也可以在分布式计算环境中实现,其中任务由通过通信网络连接的远程处理设备执行。在分布式计算环境中,程序模块可以位于包括存储器存储设备在内的本地和远程计算机存储介质中。

[0032] 参考图 1,用于实现本发明的一个示例性系统包括计算机 110 形式的通用计算设备。计算机 110 的组件可以包括,但不限于,处理单元 120、系统存储器 130 和将包括系统存储器在内的各种系统组件耦合至处理单元 120 的系统总线 121。系统总线 121 可以是若干类型的总线结构中的任一种,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中的任一种的局部总线。作为示例,而非限制,这样的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、扩展的 ISA (EISA) 总线、视频电子技术标准协会 (VESA) 局部总线 and 外围部件互连 (PCI) 总线 (也被称为 Mezzanine 总线)。

[0033] 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能够被计算机 110 访问的任何可用介质,且包括易失性和非易失性介质、可移动和不可移动介质。作为示例,而非限制,计算机可读介质可以包括计算机存储介质和通信介质。计算机存储介质包括以任何方法或技术实现的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的易失性和非易失性、可移动和不可移动介质。计算机存储介质包括,但不限于, RAM、ROM、EEPROM、闪存或其它存储器技术、CD-ROM、数字多功能盘 (DVD) 或其它光盘存储、磁带盒、磁带、磁盘存储或其它磁性存储设备、或能用于存储所需信息且可以由计算机 110 访问的任何其它介质。通信介质通常具体化为诸如载波或其它传输机制等已调制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,且包含任何信息传递介质。术语“已调制数据信号”指的是这样一种信号,其一个或多个特征以在信号中编码信息的方式被设定或更改。作为示例,而非限制,通信介质包括有线介质,诸如有线网络或直接线连接,以及无线介质,诸如声学、RF、红外线和其它无线介质。上述中任一个的组合也应包括在计算机可读介质的范围之内。

[0034] 系统存储器 130 包括易失性或非易失性存储器形式的计算机存储介质,诸如只读存储器 (ROM) 131 和随机存取存储器 (RAM) 132。基本输入 / 输出系统 133 (BIOS) 包含有助于诸如启动时在计算机 110 中元件之间传递信息的基本例程,它通常存储在 ROM 131 中。RAM 132 通常包含处理单元 120 可以立即访问和 / 或目前正在操作的数据和 / 或程序模块。作为示例,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。

[0035] 计算机 110 也可以包括其它可移动 / 不可移动、易失性 / 非易失性计算机存储介质。仅作为示例,图 1 示出了从不可移动、非易失性磁介质中读取或向其写入的硬盘驱动器 141,从可移动、非易失性磁盘 152 中读取或向其写入的磁盘驱动器 151,以及从诸如 CD ROM 或其它光学介质等可移动、非易失性光盘 156 中读取或向其写入的光盘驱动器 155。可以在示例性操作环境中使用的其它可移动 / 不可移动、易失性 / 非易失性计算机存储介质包括,但不限于,盒式磁带、闪存卡、数字多功能盘、数字录像带、固态 RAM、固态 ROM 等。硬盘驱动器 141 通常由不可移动存储器接口,诸如接口 140 连接至系统总线 121,磁盘驱动器 151 和光盘驱动器 155 通常由可移动存储器接口,诸如接口 150 连接至系统总线 121。

[0036] 以上描述和在图 1 中示出的驱动器及其相关联的计算机存储介质为计算机 110 提

供了对计算机可读指令、数据结构、程序模块和其它数据的存储。例如,在图 1 中,硬盘驱动器 141 被示为存储操作系统 144、应用程序 145、其它程序模块 146 和程序数据 147。注意,这些组件可以与操作系统 134、应用程序 135、其它程序模块 136 和程序数据 137 相同或不同。操作系统 144、应用程序 145、其它程序模块 146 和程序数据 147 在这里被标注了不同的标号是为了说明至少它们是不同的副本。用户可以通过输入设备,诸如键盘 162 和定点设备 161(通常指鼠标、跟踪球或触摸垫)向计算机 110 输入命令和信息。其它输入设备可以包括遥控器 163、麦克风、操纵杆、游戏垫、圆盘式卫星天线、扫描仪等(未全部示出)。这些和其它输入设备通常由耦合至系统总线的用户输入接口 160 连接至处理单元 120,但也可以由其它接口或总线结构,诸如并行端口、游戏端口或通用串行总线(USB)连接。监视器 191 或其它类型的显示设备(例如,电视机)也经由接口,诸如视频接口 190 连接至系统总线 121。除监视器以外,计算机也可以包括其它外围输出设备,诸如扬声器 197 和打印机 196,它们可以通过输出外围接口 195 连接。

[0037] 在某些方面,提供了笔数字化仪 165 和附属的笔或触针 166 以数字化地捕捉手画线输入。尽管示出了笔数字化仪 165 和用户输入接口 160 之间的直接连接,但在实际上,笔数字化仪 165 可以使用包括无线技术在内的任何技术,直接地、通过并行端口或其它接口以及系统总线 130 来耦合至处理器单元 110。而且,笔 166 可以含有与其相关联的照相机,以及用于将由该照相机所捕捉到的图像信息无线地发送到与总线 130 交互的接口的收发器。此外,笔可以含有除照相机之外的或代替照相机的其它传感系统,用于确定电子墨迹的笔划,这些传感系统包括加速计、磁强计和陀螺仪等。

[0038] 计算机 110 可使用至一个或多个远程计算机,诸如远程计算机 180 的逻辑连接在网络化环境中操作。远程计算机 180 可以是个人计算机、服务器、路由器、网络 PC、对等设备或其它常见网络节点,且通常包括上文相对于计算机 110 描述的许多或所有元件,尽管在图 1 中只示出存储器存储设备 181。图 1 中所示逻辑连接包括局域网(LAN) 171 和广域网(WAN) 173,但也可以包括其它网络。这样的连网环境在办公室、企业范围计算机网络、内联网和因特网中是常见的。此外,系统可以包括有线和/或无线能力。例如,网络接口 170 可以包括蓝牙、SWLan 和/或 IEEE 802.11 的组合能力类。可以理解,可以结合这些协议或替代这些协议来使用其它无线通信协议。

[0039] 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 连接至局域网 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 或用于通过诸如因特网等 WAN 173 建立通信的其它装置。调制解调器 172 可以是内置或外置的,它可以通过用户输入接口 160 或其它合适的机制连接至系统总线 121。在网络化环境中,相对于计算机 110 所描述的程序模块或其部分可以存储在远程存储器存储设备中。作为示例,而非限制,图 1 示出了远程应用程序 185 驻留在存储器设备 181 上。可以理解,所示的网络连接是示例性的,且可以使用在计算机之间建立通信链路的其它手段。

[0040] 可以理解,所示网络链接是示例性的,可以使用在计算机之间建立通信链路的其它技术。假设诸如 TCP/IP、以太网、FTP、HTTP 等各种公知协议的任一种存在,系统可以在客户机-服务器配置中操作,以允许用户从基于 web 的服务器中检索网页。各种常规网络浏览器的任何一种可以用于显示和处理网页上的数据。

[0041] 本发明的一个或多个方面可以被实施为诸如由计算机 110 等一台或多台计算机

或其它设备执行的一个或多个程序模块中的计算机可执行指令。一般而言,程序模块包括例程、程序、对象、组件、数据结构等,它们执行特定任务或实现特定抽象数据类型。计算机可执行指令可以被存储在诸如硬盘、光盘、可移动存储介质、固态存储器、RAM 等的计算机可读介质中。本领域的技术人员可以理解,程序模块的功能可以如在各个实施例中所需而结合或分布。另外,功能可以被实施为诸如集成电路、现场可编程门阵列 (FPGA) 等固件或硬件等效物的全部或部分。

[0042] 本发明的说明性实施例

[0043] 根据本发明的说明性方面,此处描述了一种文件系统,该文件系统不将项的生存周期与该项在文件系统组织结构内的位置合并,从而允许项诸如通过基于查询的属性和通过列表来参与多个组织或者根本不参与组织,这些组织构成一个或多个有向无环图 (DAG)。DAG 是一种组织,其中项可以拥有多个父项。在 DAG 中,项不能够是其祖先的父项,否则会形成环。

[0044] 此处描述的文件系统可以被实施化为存储在计算机可读介质上的计算机可执行指令,例如,文件系统 138 和 / 或文件系统 148 (图 1)。另外,图 1 中的其它计算机可读介质或存储器也可以包括与本发明所述的文件系统相同或不同的文件系统。例如,介质 152 和 156 也可以包括文件系统。为说明起见,此处将参考文件系统 148 来描述文件系统。

[0045] 参考图 3 和图 4,文件系统 148 主要依赖于两张表——位置表 (LOC) 301 和组织表 (ORG) 401。位置表 301 也被称为 LOC 表,它存储关于存储在文件系统 418 中的每一项的物理和 / 或逻辑位置的信息,包括,例如,文件标识符 (ID) 303、文件名 305、存储卷 307、位置 309 以及文件大小 311。文件 ID 303 是用来引用每一项的主键或引用,且对存储在文件系统 418 中的每一项是唯一的。文件 ID 可以包括数字、字母文本、符号或者任何组合,只要每一文件 ID 唯一地标识存储在文件系统 148 中的单个项。

[0046] 文件名 305 可以包括由文件系统 418 的用户提供的关于项的短的描述性文件名。文件名 305 较佳地是用户对每一文件的主要概念参考,且可以由用户以描述性的格式指定,使得文件可以由用户基于文件名来容易地识别。文件名 305 可以是有限长度的,例如 8.3 格式,最长 256 个字符,或是某些其它最大值。较佳地,对文件名彼此是否可以重复,或者甚至是否要求文件名或已经分配文件名没有限制。即,多个文件可以各自被命名为“README.TXT”或“Read this first”,即使它们处于相同的目录、文件夹或列表中,只要每一文件拥有唯一的文件 ID 303。这不像要求每一文件在其父项的上下文中拥有唯一的名称的之前的文件系统。

[0047] 卷 307、位置 309 和大小 311 可以用来从物理或逻辑存储位置中检索项。卷 307 可以指的是物理或逻辑存储设备,例如,物理或逻辑硬盘驱动器、光盘驱动器、存储介质卡、固态存储设备等。位置 309 可以指的是项在标识的物理或逻辑卷上的实际起始位置,而大小 311 可以指的是其中项以标识的位置开始被存储在所标识的卷上的空间量的定量测量。位置表 301 中可以包括另外的或替换的信息,只要包含的信息可用于从项的各自的标识的存储位置检索每一存储的项。

[0048] 组织表 401 也被称为 ORG 表,它存储了存储在文件系统 418 中的项之间的层次关系。如上简要所述,为概念上在文件系统 148 中排列项,ORG 表 401 可以定义包含一棵或多棵树的一个或多个有向无环图 (DAG)。ORG 表 401 可以对每一关系包括父项 403 以及子项

405。图 5 示出了如由 LOC 表 301 的各部分定义的以及如由 ORG 表 401 定义的 DAG 501 的示例。DAG 501 不是树,因为项 13 的父项是“To Do List(要完成的事情列表)”和“Items For My Trip(关于我的旅行的项)”(也被称为“在两者中”),项 14 的父项是“Items For My Trip”和“Key Slide Decks(主要滑板)”。文件系统 418 中的每一项可以没有父项、有一个父项或者有两个或更多个父项,只要组织结构仍旧是 DAG。

[0049] 图 6 示出了可以在文件系统 148 中使用的软件模块的框图。文件系统 148 可以含有管理文件系统 148 的总体操作的文件系统管理器程序模块 601。管理器 601 可以使用此处所述的 LOC 表 301 和 ORG 表 401 来添加、更新、删除和读取 / 查询存储在数据存储 611 中的项,同时实施关于施加在 DAG 上的任何约束(例如,不允许环)的规则。管理器 601 可以展示一个或多个应用程序编程接口(API),通过这些 API,诸如操作系统等更高级程序可以与文件系统 148 交互。API 可以包括 Add_Item API 603、Update_Item API 605、Delete_Item API 607 以及 Query_Item API 609。

[0050] Add_Item API 603 可以由操作系统或其它更高级程序或应用程序在有新的项将被存储在文件系统 148 中时调用。Add_Item API 603 可以接受指向该新的项的当前临时存储位置的指针,以及文件名和可任选的预期的存储卷和 / 或父项 ID 作为输入。Add_Item API 603 可以任选地为新添加的项返回文件 ID。

[0051] Update_Item API 605 可以被调用来更新已经存储在文件系统 148 中的现有项,且可以接受文件 ID 和从指向该项的更新版本的指针、ORG 表 401 中的内容的更新的关系和 / 或更新的文件名中选出的一个或多个可任选地包含的信息作为输入。

[0052] Delete_Item API 607 可以被调用以通过向 Delete_Item API 传递将要删除的文件 ID 来从文件系统 148 中移除项。仅当明确地调用 Delete_Item API 607 时,才从文件系统 148 中删除项。

[0053] Query_Item API 609 可以在需要从文件系统中检索项以供 OS 或应用程序使用时调用。Query_Item 接受文件 ID 作为输入,并返回请求的项。以上列出的 API 代表了较佳地包含在本发明的说明性实施例中的 API 类型。文件系统 148 可以基于系统需求和设计包含另外的或替换的 API。

[0054] 因为文件系统 148 允许每一项拥有零个或多个父项,因此在文件系统的单个卷中可能存在多个独立的数据结构(即,多个 DAG)。这不像一般将客户机(用户或应用程序)限于对每个卷在单棵树中存储项的已知文件系统。这样,如此处所述的,其中项被存储在文件系统 148 中的概念空间被称为文件区域(FR)。文件区域指的是用于控制项的生存周期的组织区域的高级用户概念。只要项仍旧在文件区域内,该项将不被删除,而不考虑该项是否服从 ORG 表 401 中定义的任何父-子关系。

[0055] 文件区域可以在概念上被看作其中可以放置项的盒子,而不考虑项是否在概念上是相关的。如果项被放置在文件区域中,那么项保持处于文件区域中,直到用户删除该项。尽管项是在文件区域中,但是项可以在文件区域内以一个或多个 DAG 中的任何一个来组织,而对该项是否在文件区域中没有影响,且对该项的生存周期没有影响。即,该文件不会被删除,直到用户肯定地指示文件系统删除该文件。文件区域可以包括任何任意定义的存储区域,而不考虑所有包含的数据存储是否彼此在同一计算机、网络等中是物理上邻近的,只要数据存储是作为单个文件区域来管理以确保项的生存周期。例如,文件区域可以包括

计算机上的所有内部硬盘驱动器、计算机上的单个硬盘驱动器的一部分、单独的或与本地存储结合的网络化存储、或者任何其它定义的存储空间。

[0056] 图 7 和 8 示出了含有三个独立 DAG 501、703 和 705 的文件区域 701。在 DAG 501、703 和 705 之间的唯一强制关系是它们位于同一文件区域内，它们之间不存在必须的概念关系。图 8 示出了描述图 7 中所示的 DAG 的 ORG 表 801。图 7 示出了文件区域 701 的概念视图，而图 8 示出了在文件区域 701 中使用的持有链接。文件区域 701 可以由用户打开，此时文件系统 148 可以向用户呈现由文件系统定义的文件区域中的所有“顶层”或“浮动”项。如此处所使用的，如果项不是来自对应于该文件区域的 ORG 表中的其它项的任何持有链接的目标，则项被定义为“顶层”或者“浮动的”。这样，在图 7 中，项 i1, i07, “ToDo List” 和 “Items For My Trip” 被认为是顶层项，即浮动项。对终端用户，列表 i7 是组织数据结构形式的。基于多个父项的概念，项可以处于多个列表中，例如项 i3。列表是持有其它项、已排序的或未排序的项，且列表可以代替用户在使用以往的文件系统时惯用的文件夹 / 目录。列表也可以在其它列表中或持有其它列表。

[0057] 文件系统 148 在用户或应用程序（共同被称为客户机）何时想要从项的基础组织结构中移除该项，以及客户机何时想要将该项从文件系统中完全删除之间进行区分。图 9 和 10 示出了在用户或应用程序从 Key Slide Decks 列表中移除项 i5 之后的文件区域 801。即，管理器 601 基于通过 Update_Item API 605 的请求，移除 ORG 表 1001 中将 Key Slide Decks（文件 ID 000007）标识为项 i5（文件 ID 000006）的父项的持有链接。项 i5 不再是多项组织结构的一部分，因为关于项 i5 的所有持有链接都已经从 ORG 表 401 中移除。然而，因为项 i5 仍旧保持在文件区域 801 中，所以不从文件系统 148 或 LOC 表 301 中移除项 i5。换言之，因为客户机没有请求删除项 i5，而是请求移除项 i5，所以不从数据存储器 611 和 LOC 表 301 中删除项 i5。仅当客户机通过 Delet_Item API 肯定地选择项 i5 并请求删除操作之后，管理器 601 才会从数据存储器 611 和 LOC 表 301 中移除项 i5（文件 ID 000006）。

[0058] 参考图 11 和 12，文件系统 148 可用于基于在例如目录树等先前已知的文件系统中使用的组织结构来模拟用户熟悉的组织结构。回头参考 LOC 表 301（图 3），具有文件 ID 000301-000312 的项表示对存储在文件系统 148 中的列表的持有链接。每一未排序的列表表示在文件区域 1201（图 12）和 ORG 表 1101（图 11）中概念上表示的目录树 1203 中的目录。目录树 1203 和先前已知的目录树的主要区别在于，“Shared（共享的）”子目录可通过两个“User Data（用户数据）”未排序列表中的任一个来访问，这样利用了文件系统 148 的多父项能力。以这种方式，用户之间可以容易地共享目录，同时仍旧向用户提供他们熟悉的组织结构。

[0059] 根据本发明的说明性实施例，文件系统 148 向计算机系统或网络的每一单独用户提供在其上实现文件系统 148 的私有工作空间，并向该计算机系统或网络的所有用户提供可用的公共共享工作空间。形成名字空间，使得它具有 UserData 下的 Private（私有）和 Shared 文件区域。这允许用户针对系统的查询可以作为由 User Data 形成的项域的根，从而将在所有典型查询中返回私有和共享的项。使用图 12 的名字空间，当用户与文件系统中的项交互时（获得新的相片或音乐、创建文档等），用户仅需在概念上考虑问题“这是私有的还是共享的？”并将项放置于合适的工作空间中。共享工作空间的所有用户然后拥有对共享工作空间的公共可见性。另外，每个工作空间可以有共享的回收站，这样如果一个用户

删除了另一个用户仍旧想要的项,那么第二个用户可以找回它。

[0060] 对公共共享工作空间的情形不限于家庭用户。在公司域上,知识工作者可以为同事建立共享的工作空间以在项目上协作。工作空间的灵活性可以向工作空间的每一用户提供组织项、在工作空间内查询项以及提供对这些项的公共总体可见性的能力。

[0061] 图 13 示出了用于从项的底层组织结构中移除项和 / 或从文件系统 148 中删除项的方法。最初,在步骤 1301 中,客户机选择具有特定文件 ID 的项。下一步,在步骤 1303 中,客户机指示该客户机想要将该项从文件系统 148 中完全地删除,或者该客户机仅想从项的底层组织结构中移除该项。如果客户机在步骤 1303 中选择“移除”,那么在步骤 1305 中,文件系统管理器从 ORG 表中移除其中该项的文件 ID 在子项栏中出现的任何条目,然后该方法结束。

[0062] 如果客户机在步骤 1303 中选择“删除”,那么在步骤 1307 中,对其中该项的文件 ID 出现在父项栏中的每一 ORG 表条目,文件系统管理器 601 在 ORG 表中添加其中子项是 child 而该项的父项是 parent 的条目(如果该条目不是已经存在于 ORG 表中)。可以使用的伪代码如下:

[0063] 对其中 $X_{\text{parent}} = \text{file_ID}(\text{ITEM})$ 的每一 ORG 表条目 X

[0064] 对其中 $Y_{\text{child}} = \text{file_ID}(\text{ITEM})$ 的每一 ORG 表条目 Y

[0065] 如果 ORG 表不含有条目 $(Y_{\text{parent}}, X_{\text{child}})$

[0066] 那么添加 ORG 表条目 $(Y_{\text{parent}}, X_{\text{child}})$

[0067] 在相应地更新 ORG 表之后,文件系统管理器在步骤 1309 中从 ORG 表中删除引用该项的文件 ID 作为父项或者子项的任何条目。最后,在步骤 1311 中,文件系统管理器 601 从 LOC 表中删除指示该项存储在哪里的条目。文件系统管理器 601 还可以执行安全擦除操作,并使用垃圾数据重写存储在该项位置上的数据,使得不能执行恢复删除操作。

[0068] 或者,文件系统管理器可以跳过步骤 1307 而在步骤 1309 中仅从 ORG 表中移除其中该项要么是父项要么是子项的任何条目。该替换实施方式的副作用是,被删除的项的每一子项被从文件系统 148 的组织结构中移除,除非该子项是以仍旧保持在文件系统 148 的组织结构中的另一项作为父项的。

[0069] 示出上述方法,使得所选择的项和该项指向的任何项被从底层组织结构中移除。然而,本领域的技术人员可以理解,可以使用其它方法,其中当从组织结构中移除项时,不将该项的子项也从组织结构中移除。移除方法是如何执行的对于以下事实是次要的,即移除和删除是两个独立的过程:删除过程从存储中完全删除项,而移除过程从底层组织结构中移除项而不影响该项的生存周期(即,不从存储或文件系统中删除该项)。

[0070] 传统文件系统导航工具,例如由华盛顿州雷德蒙市的微软公司推出的 Windows EXPLORER 系统导航工具,一般仅显示文件系统组织结构中的文件。这样,如果客户机从文件系统 148 的组织结构中移除了项,即该项不存在于 ORG 表中,那么该项将不会在文件系统导航显示中出现。然而,使用本发明的各方面,客户机可以向文件系统管理器 601 查询所有浮动或顶层项(即,没有父项的项),以找出在组织结构内不存在的任何项。

[0071] 文件系统 148 可以基于由 ORG 表 401 定义的组织结构提供附加的特征,诸如查询域和安全性。即,除了定义组织结构之外,ORG 表中的持有链接也可以用于形成可查询项域,或者也被称为名字空间。客户机可以查询涉及到该项是父项的至少一条持有链接的任

何项。例如,参考图 7,对“Items For My Trip”的查询返回项 i3、i4、i5 和 i8,以及 Key Slide Decks。重复的项较佳地仅返回一次。例如,Key Slide Decks 也是 i4 的父项,它较佳地仅被返回一次。

[0072] 文件系统 148 的组织结构也可以用于传播名字空间以供传统应用程序使用,也可以向新用户提供概念上可识别的用户界面来与文件系统 148 交互。例如,参考图 12,“U1”下的项“User Data”可由客户机如“\Users\U1\User Data”来引用,这样向终端用户和传统应用程序提供概念上熟悉的界面。

[0073] 由 ORG 401 定义的组织结构也可以用于在文件系统 148 中传播安全信息。即,安全表(未示出)可以提供关于项的安全信息。默认地,如果用户能够访问给定项,该用户默认地不论直接或间接地也能够访问该给定项指向的所有其它项。

[0074] 例如,项“U1”(图 12)的安全性可以指示只有用户名= Ross 的用户才能访问 U1 中的文件。如果没有提供了父项为 U1 的项“App Data(应用程序数据)”和“User Data”的不同安全信息,那么也仅有 Ross 才能查看这些项。

[0075] 对“Private”(由父项为“U1”的“User Data”作为父亲)和“Shared”,这也成立。项“U2”(图 12)的安全性可以指示只有用户名= Jordan 的用户才能访问 U2 中的文件。如果没有提供父项为“U2”的项“Application Data”和“UserData”的不同安全信息,那么也仅有 Jordan 才能查看这些项。对“Private”(由父项为“U2”的“User Data”作为父亲)和“Shared”,这也成立。尽管看上去项“Shared”具有不一致的安全信息,安全信息较佳地是加性的。这样,项“Shared”将会接收到来自“U1”和“U2”的安全信息,以允许“仅 Ross 和 Jordan”。或者,假设项“U1”具有等价于“仅 Ross,而不是 Tom”的许可。项“Shared”则将接收到等价于“仅 Ross 和 Jordan,而不是 Tom”的安全许可。

[0076] 当从 ORG 中移除项,安全性停止传播至被移除的项。例如,如果如图 9 中所示,从“Key Slide Decks”中移除项 i5,那么用户不再拥有访问 i5 的能力,除非该用户被给予访问 i5 的许可。然而,文件系统 148 可以指定默认地客户机可以访问所有项,除非该客户机被明确地禁止访问项。

[0077] 因为列表也形成如上所述的名字空间,因此如果来自 Windows XP 或其它传统机器(可能不配备如此处所述的文件区域、DAG 或列表来工作)的用户连接至文件系统 148,用户可以见到如可以被导航的文件夹可见的列表。类似地,当用户在列表(或者在传统操作系统或者能够使用文件系统 148 的系统上)的上下文内的给定项上双击时,那么文件系统向请求应用程序提供由当前列表形成的名字空间。这向应用程序提供了必要的上下文。例如,如果用户试图通过列表“To Do List”来打开项 i2,那么文件系统将向应用程序返回通过“To doList”形成的路径,即“\To do List\i2”。这样,如果用户选择在应用程序内使用命令“File|Save As...”(例如,用于创建该项的新副本),那么应用程序将具有正确的上下文来提供给用户。

[0078] 尽管本发明是相对于包括实现本发明的当前最佳的方式的特定示例来描述的,但本领域的技术人员可以理解,存在上述系统和技术的众多变化和改变。这样,本发明的精神和范围应该如所附权利要求所述来宽泛地解释。

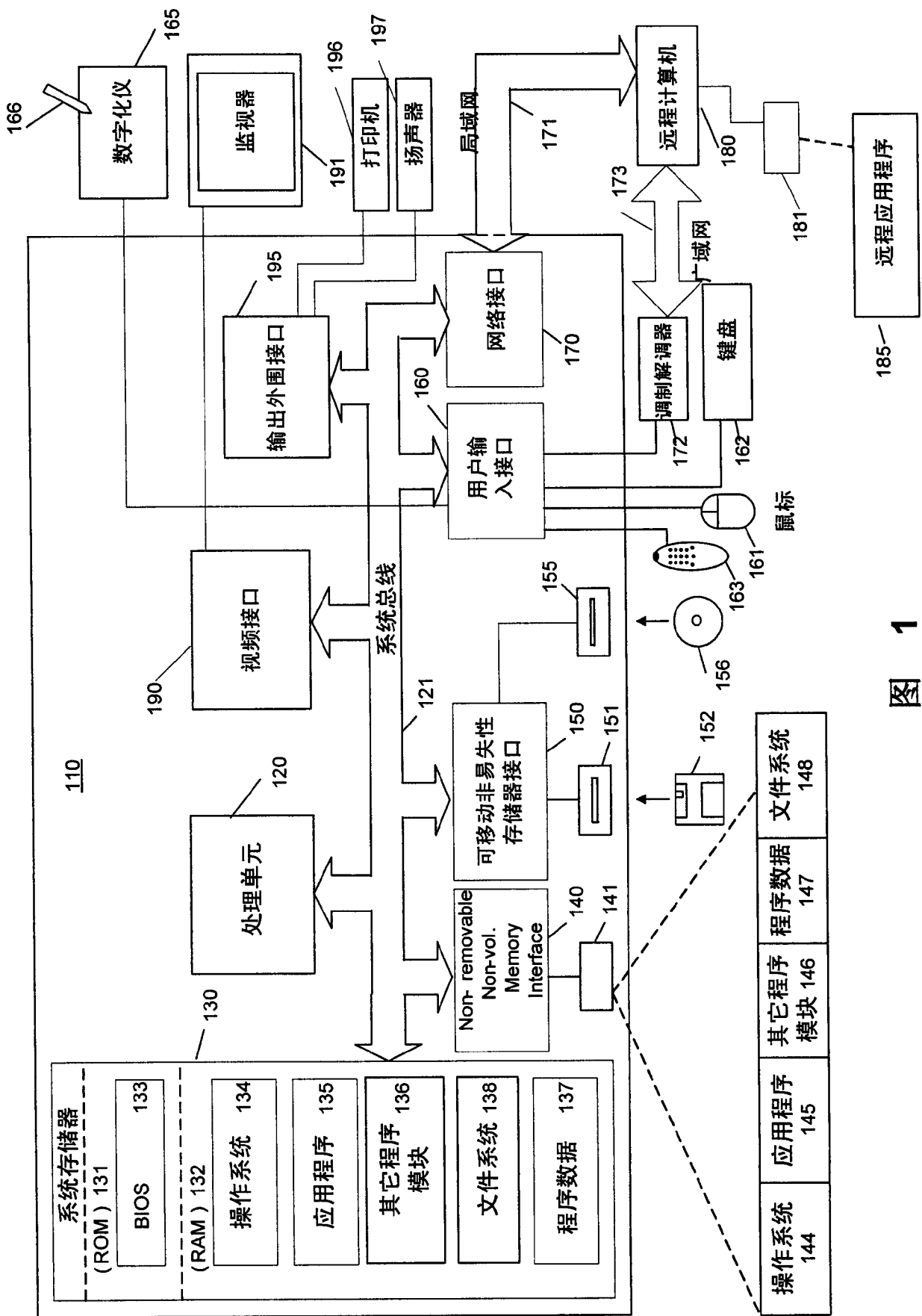


图 1

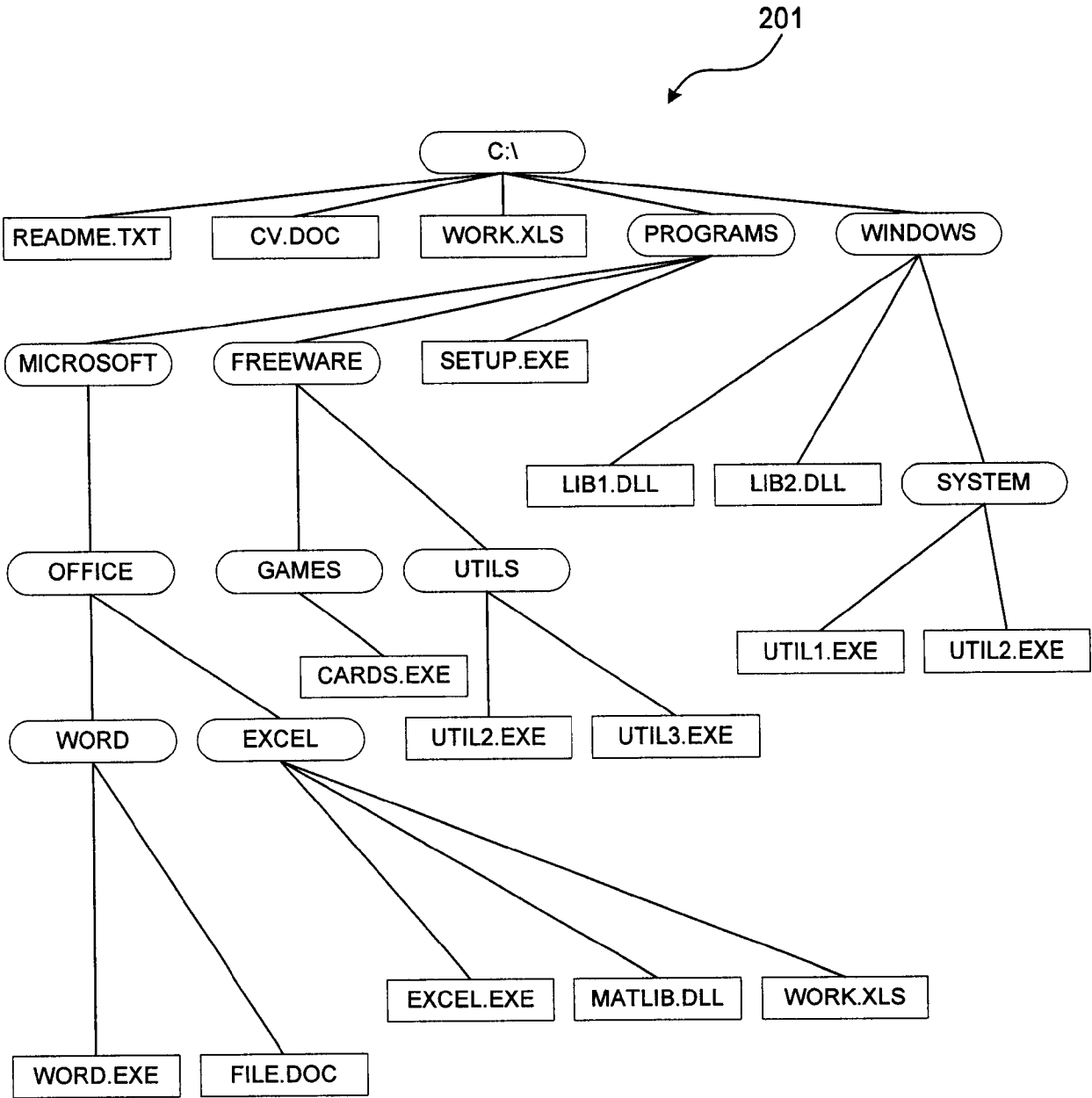


图 2

现有技术

文件 ID	文件名	卷	位置	大小
000001	To Do List	C	19836	2
000002	I2	C	19838	20
000003	I3	C	19858	14
000004	Items For My Trip	C	02864	3
000005	I4	D	34884	32
000006	I5	C	07893	18
000007	Key Slide Decks	D	02389	2
000008	I8	D	00034	97
000009	I9	C	00625	23
...	...	...	...	...
00085	I1	C	08762	13
...	...	...	...	...
00135	I2	D	09238	27
...	...	...	...	...
00245	L1	C	04874	4
00246	I7	L	07629	45
...	...	...	...	...
00301	\	C	05672	1
00302	Users	C	05673	1
00303	Machine	C	05674	1
00304	U1	C	05675	1
00305	U2	C	05676	1
00306	App Data	C	05677	1
00307	User Data	C	05678	1
00308	App Data	C	05679	1
00309	User Data	C	05680	1
00310	Private	C	05681	1
00311	Private	C	05682	1
00312	Shared	C	05683	1

图 3

父	子
000001	000002
000001	000003
000004	000003
000004	000005
000004	000007
000004	000008
000007	000005
000007	000006

图 4

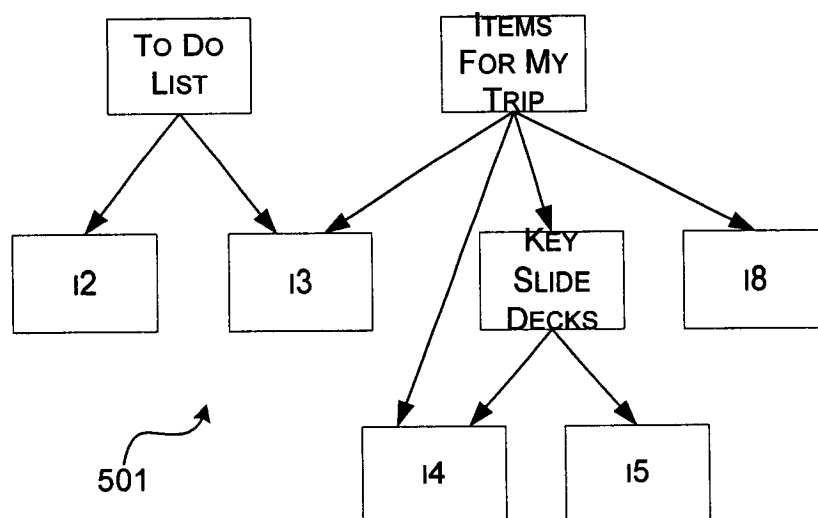


图 5

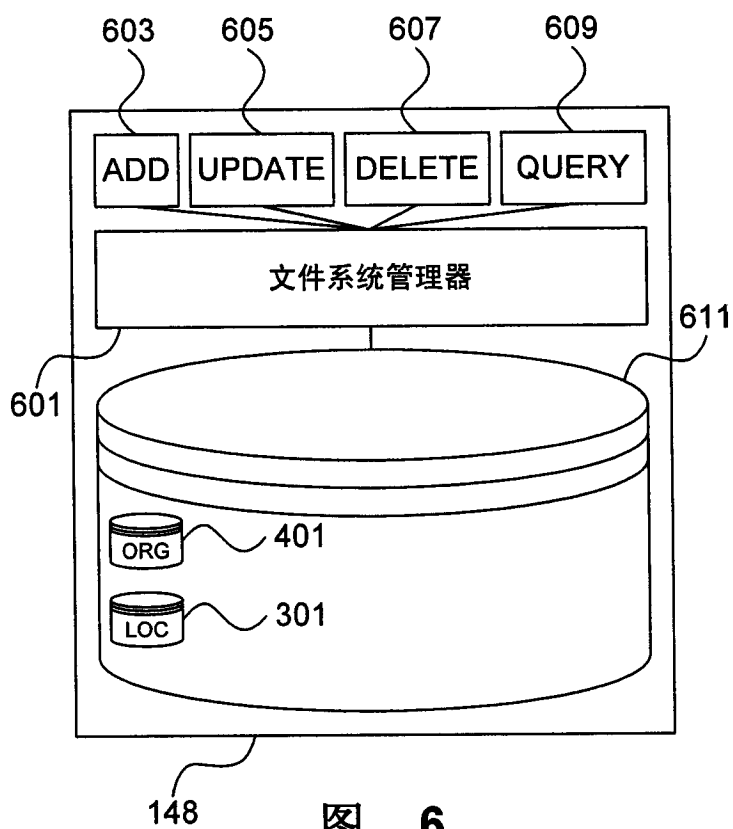
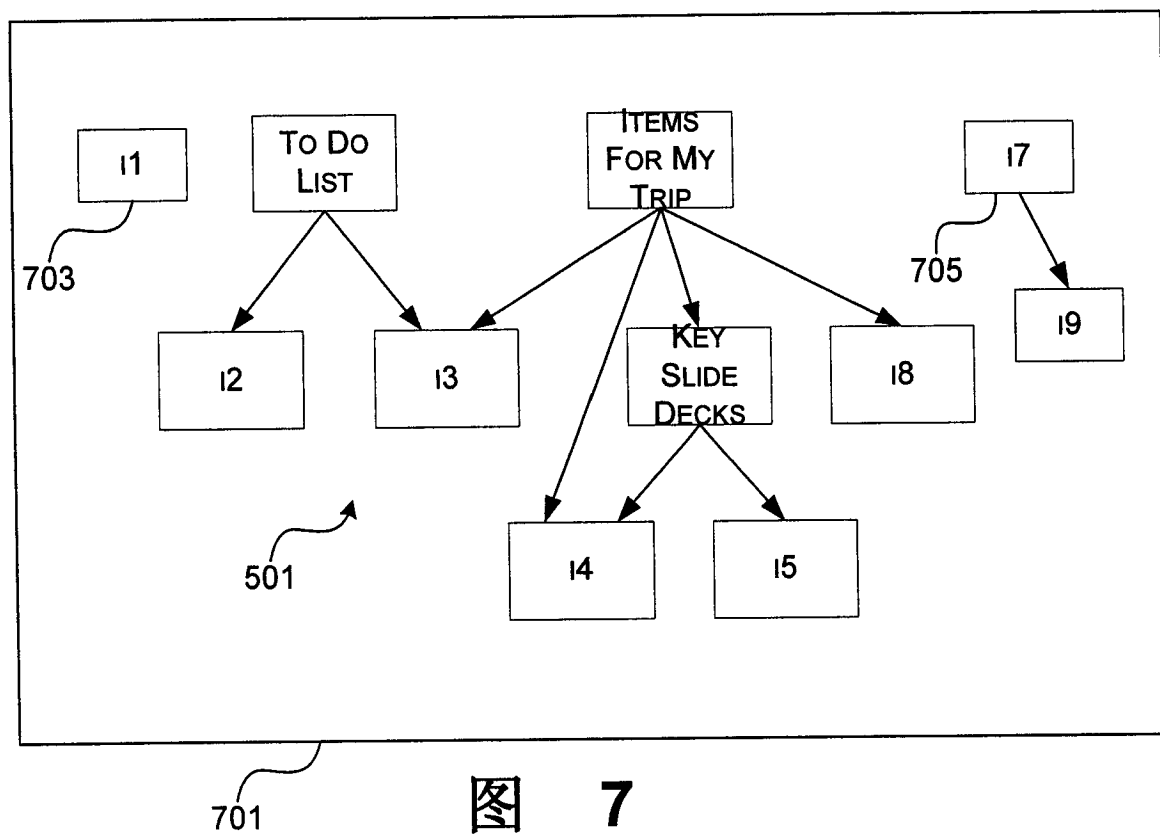


图 6



801

父	子
000001	000002
000001	000003
000004	000003
000004	000005
000004	000007
000004	000008
000007	000005
000007	000006
000246	000009

图 8

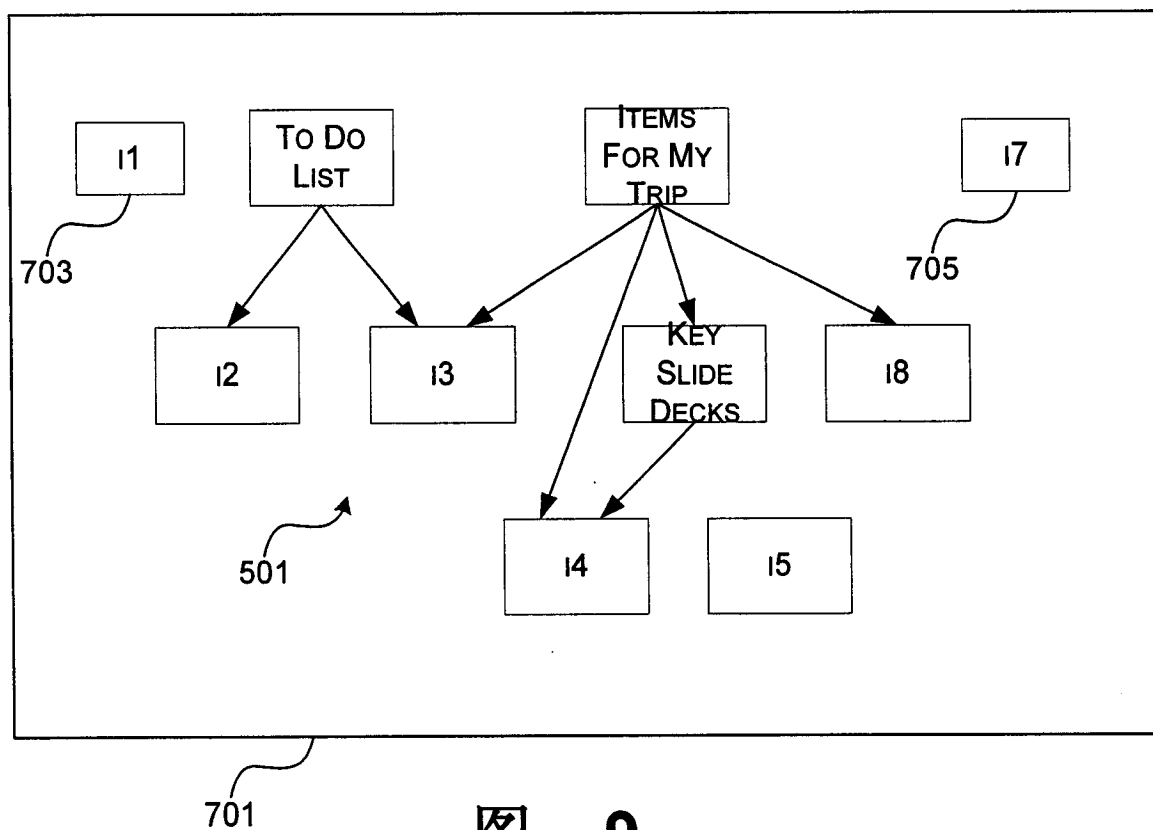


图 9

1001

父	子
000001	000002
000001	000003
000004	000003
000004	000005
000004	000007
000004	000008
000007	000005
000246	000009

图 10

1101

父	子
00301	00302
00301	00303
00302	00304
00302	00305
00304	00306
00304	00307
00305	00308
00305	00309
00307	00310
00307	00312
00309	00311
00309	00312

图 11

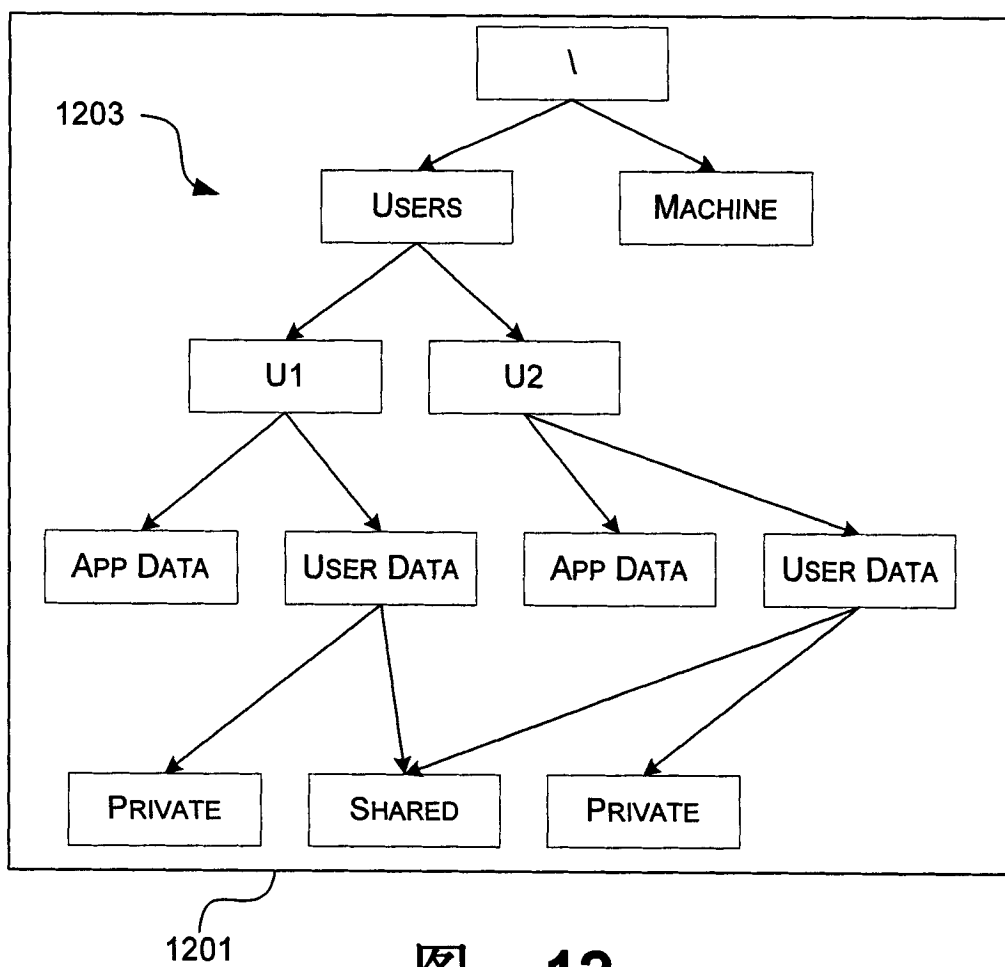


图 12

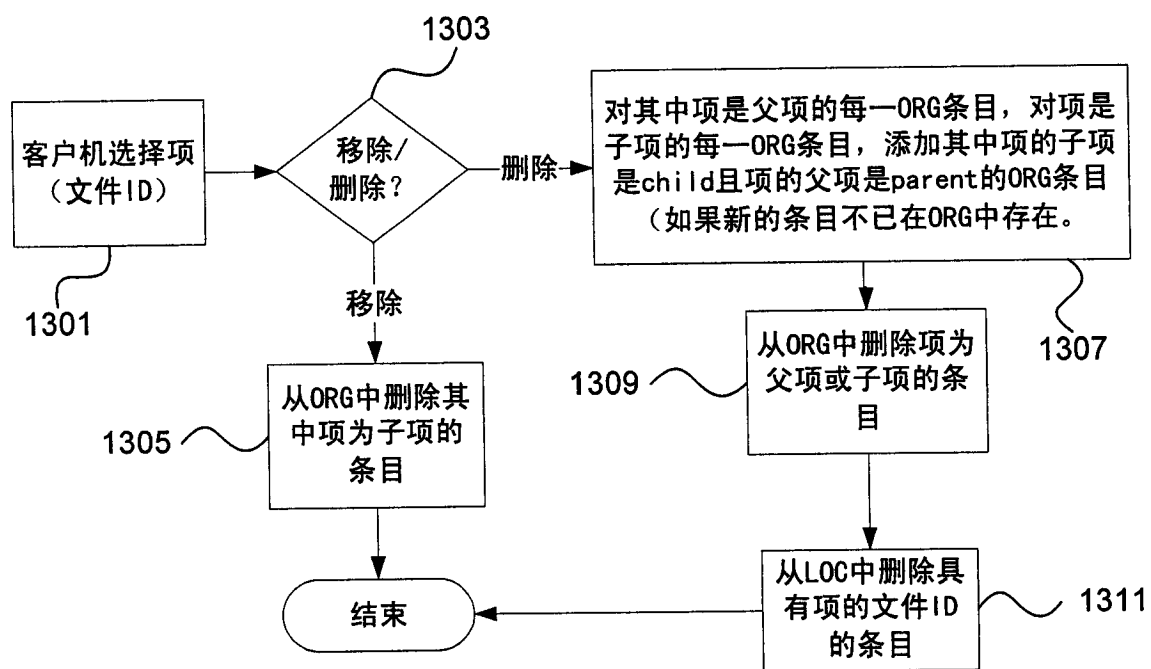


图 13