[19] 中华人民共和国国家知识产权局

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

[12] 发明专利申请公布说明书

[21] 申请号 200880010343.9

[43] 公开日 2010 年 3 月 10 日

[11] 公开号 CN 101669118A

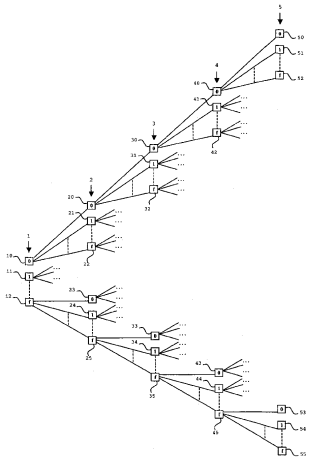
[22] 申请日 2008.2.12
[21] 申请号 200880010343.9
[30] 优先权
[32] 2007. 2.13 [33] FR [31] 0753222
[86] 国际申请 PCT/FR2008/000177 2008.2.12
[87] 国际公布 WO2008/113921 法 2008.9.25
[85] 进入国家阶段日期 2009.9.28
[71] 申请人 STG 交互公司
地址 法国巴黎
[72] 发明人 亚历克西斯·塔马斯
阿莫里·格兰贝尔
[74] 专利代理机构 北京集佳知识产权代理有限公司
代理人 李春晖 陈 炜
权利要求书 3 页 说明书 14 页 附图 9 页

[54] 发明名称

文件管理方法

[57] 摘要

本发明涉及一种文件管理方法，包括组织数据文件库的第一步骤，该第一步骤包括创建 M 层目录的树形图，每个目录包含 N 个目录，M 是大于 1 的整数，该方法还包括记录数据文件的记录步骤，该记录步骤包括：对待记录的数据文件 F_i 的识别码应用哈希函数，根据前一步骤的结果在树形图中确定目标多层目录 R_{di} 的路径，在由所述哈希函数确定的所述目录 R_{di} 中在由数据文件的识别码决定的位置记录数据文件，该方法还包括读取数据文件的读取步骤，该读取步骤包括：对待读取的数据文件 F_j 的识别码应用同一哈希函数，根据前一步骤的结果在树形图中确定目标目录 R_{cj} 的路径，在由所述哈希函数确定的所述目录 R_{cj} 中在由数据文件的识别码决定的位置读取数据文件。



1. 一种文件管理方法，包括组织数据文件库的第一步骤，所述第一步骤包括创建M层目录的树形图，每个目录包含N个目录，M是大于1的整数，该方法还包括记录所述数据文件的记录步骤，所述记录步骤包括：

－对待记录的数据文件 F_i 的识别码应用哈希函数，

－根据前一步骤的结果，在具有多个层的所述树形图中，确定目标目录 R_{di} 的路径，

－在由所述哈希函数确定的所述目录 R_{di} 中，在由所述数据文件的识别码决定的位置，记录所述数据文件，

该方法还包括读取数据文件的读取步骤，所述读取步骤包括：

－对待读取的数据文件 F_j 的识别码应用同一哈希函数，

－根据前一步骤的结果，在所述树形图中确定目标目录 R_{cj} 的路径，

－在由所述哈希函数确定的所述目录 R_{cj} 中，在由所述数据文件的识别码决定的位置，读取所述数据文件。

2. 根据前述权利要求所述的文件管理方法，其特征在于，所述数据文件分布在Q个存储单元中，每个存储单元对应于P层目录，每个目录包含N个目录。

3. 根据前述权利要求中任一项所述的文件管理方法，其特征在于，N等于16，以及所述哈希函数是SHA1函数。

4. 根据前述权利要求中任一项所述的文件管理方法，其特征在于，每个数据文件都包括头部和主体，所述主体包括文件的可修改内容，所述主体前面置有头部，该头部包括所述文件在文件管理系统中的状态参数，所述方法包括通过对一系列操作的处理来获取目标文件的预先获取步骤，该获取引起所述头部的参数中至少一个的状态改变，所述状态改变阻止另一处理进行获取，直到释放目标文件。

5. 根据前述权利要求中任一项所述的文件管理方法，其特征在于，所述获取、读取和释放操作由客户软件控制，相应的步骤由服务器软件执行。

6. 根据前述权利要求中任一项所述的文件管理方法，其特征在于，包括获取现有数据文件的步骤，该步骤采用基于使用名称而计算出的文件

名称作为输入(150),所述步骤包括第一任务(100),所述第一任务(100)包括尝试在操作系统的文件系统层次上同时打开和锁定所述文件,通过要求打开和锁定所述文件阻止后面的处理打开和锁定所述文件,直到关闭和解锁所述文件。

7. 根据前述权利要求中任一项所述的文件管理方法,其特征在于,包括获取新的数据文件的步骤,该步骤采用基于使用名称而计算出的文件名称作为输入(250),所述第一任务(200)包括尝试在操作系统的文件系统层次上同时创建、打开和锁定新的文件,通过要求创建、打开和锁定所述文件阻止后面的处理打开和锁定所述文件,直到关闭和解锁所述文件。

8. 根据前述权利要求中任一项所述的文件管理方法,其特征在于,包括获取现有数据文件,若所述文件不存在则创建所述文件的步骤,所述功能采用基于使用名称而计算出的文件名称作为输入(350),所述第一任务(300)包括尝试在操作系统的文件系统层次上同时打开和锁定文件,如果所述文件不存在则创建所述文件,通过要求打开和锁定文件阻止后面的处理打开和锁定所述文件,直到关闭和解锁所述文件。

9. 根据前述权利要求中任一项所述的文件管理方法,其特征在于,包括在不修改数据的情况下释放获取到的数据文件的步骤,所述功能采用如前所述基于使用名称而计算出的文件名称以及在前面获取数据文件时送回的获取识别码作为输入(450),所述第一任务(400)包括尝试在操作系统的文件系统层次上同时打开和锁定所述文件,通过要求打开和锁定文件阻止后面的处理打开和锁定所述文件,直到关闭和解锁所述文件。

10. 根据前述权利要求中任一项所述的文件管理方法,其特征在于,包括在修改数据的情况下释放获取到的数据文件的步骤,所述功能采用如前所述基于使用名称而计算出的文件名称、在前面获取数据文件时送回的获取识别码以及所述文件的可修改内容作为输入(550),所述第一任务(500)包括尝试在操作系统的文件系统层次上同时打开和锁定所述文件,通过要求打开和锁定所述文件阻止后面的处理打开和锁定所述文件,直到关闭和解锁所述文件。

11. 根据前述权利要求中任一项所述的文件管理方法,其特征在于,包括在删除获取到的数据文件的情况下释放所述数据文件的步骤,所述功能采用如前所述基于使用名称而计算出的文件名称以及在前面获取数据文件时送回的获取识别码作为输入(650),所述第一任务(600)包括尝

试在操作系统的文件系统层次上同时打开和锁定所述文件，通过要求打开和锁定所述文件阻止后面的处理打开和锁定所述文件，直到关闭和解锁所述文件。

12. 根据前述权利要求中任一项所述的文件管理方法，其特征在于，包括在不获取现有数据文件的情况下简单读取所述文件的步骤，所述功能采用如前所述基于使用名称而计算出的文件名称作为输入（750），所述第一任务（700）包括尝试在操作系统的文件系统层次上同时打开和锁定所述文件，通过要求打开和锁定所述文件阻止后面的处理打开和锁定所述文件，直到关闭和解锁所述文件。

文件管理方法

本发明涉及数字文件管理领域，尤其涉及在记录介质上对数字文件的记录、读取和修改。更确切地，本发明涉及与电脑操作系统（operating system）所专有的文件系统（file system）相互作用的数据文件管理方法和系统。

通常，根据存储器空间的可用性，将新的数据文件记录在存储器（例如硬盘）的位置中。在用硬盘进行记录的情况下，硬盘在格式化时被组织成初始化的块。这些块在格式化时被分配到存储器的物理扇区上。

计算机上的数据文件管理是通过计算机操作系统的文件系统来实现的。

操作系统的文件系统使用记录介质上的一个或多个分区。分区包括用于记录任意数据文件的存储空间，以及记录存储空间的特征、尤其是文件在存储空间中的地址的索引空间。

某些文件构成目录，将与数据文件和属于同一分区的目录相关的特征的列表集中在一起。

对于存储容量较大（例如几千兆字节）的记录介质，当要被记录、读取或修改的数据文件的数量在同一目录中变得较大时，文件系统的索引系统的性能恶化。实际上，目录类文件的组织包括含有大量待处理事件的列表，这在数据文件的数量达到数万时会以造成严重阻碍的方式增加存取文件的时间。在数据文件必须重命名或删除时引起的目录类文件的碎片化现象加剧了所述性能恶化。

通过 Linux EXT3 文件系统已知的另一组织方式，提出了一种目录类文件的组织，包括 B－树类型的内部索引系统，以及针对每个数据文件，基于哈希函数（fonction de hachage）的结果来计算数据文件名的摘要，以便在含有大量数据文件的目录中加速搜索。该解决方案允许在同一目录中操作十多万数据文件。然而，超过某一阈值，哈希函数的极限使得不再能够避免引起为不同名字的两个数据文件分配相同摘要的冲突。对于 EXT3 文件系统，对于 Linux 2.6 内核版本，一旦同一目录中的数据文件的数量达到约 500000，该极限出现。

在现有技术中已知专利 US5742817，描述了一种文件服务器，提出了目的在于简化寻址处理的文件查询方法。该文献提出通过由文件管理系统自动分配的 INODE 识别处理来加速存取文件。该处理包括从该识别码中提取文件系统中的子目录所对应的三组十六进制数据。

美国专利申请 US2004/0236761 描述了将文件记录在一系列目录和子目录中的系统。在树形图的最后一层，可进行包括以下步骤的处理：根据文件名计算摘要以确定记录了所述文件的“HASH SLOT（哈希时隙）”子目录。

换言之，以传统的树形图形式组织存储器空间，在树形图的最后一层，一个处理在单一层次的集合中确定子目录。

该方案限制了可以被记录的文件的数量。文件管理系统的寻址方式限制了子目录的数量，例如对于 UNIX 类型的管理系统为 65536。

此外，上述各子目录内部的文件的数量也由同样的界限所限制。

总之，容量受到限制，为了充分使用容量，必须使子目录的数量最大化。而在单一“耙形”层子目录中，子目录数量限制了文件管理系统的性能。

本发明的目的在于通过提出一种稳定耐用而快速的文件管理方法来克服该缺点，从而允许操作的数据文件的数量仅受可用的存储空间限制，而不受由文件系统对数据文件名称执行的处理的限制。因此根据本发明的方案不限制使用大容量存储器的能力，这是由于唯一的限制是这些存储器的物理容量，而不再是文件系统要求的操作模式。对于例如文件服务器应用，该方案允许管理很大数量的数据文件。

根据最广泛的含义，本发明涉及一种文件管理方法，包括组织数据文件库的第一步骤，该第一步骤包括创建 M 层目录的树形图，每个目录包含 N 个目录，M 是大于 1 的整数，该方法还包括记录数据文件的记录步骤，该记录步骤包括：

－对待记录的数据文件 F_i 的识别码应用哈希函数，

－根据前一步骤的结果，在具有多个层的树形图中，确定目标目录 R_{di} 的路径，

－在由所述哈希函数确定的所述目录 R_{di} 中，在由数据文件的识别码决定的位置，记录数据文件，

该方法还包括读取数据文件的读取步骤，该读取步骤包括：

－对待读取的数据文件 F_j 的识别码应用同一哈希函数，

－根据前一步骤的结果，在树形图中确定目标目录 R_{cj} 的路径，

－在由所述哈希函数确定的所述目录 R_{cj} 中，在由数据文件的识别码决定的位置，读取数据文件。

为了并行使用存储单元并改进性能（存取速度和/或存储器容量），根据一个具体实施方式，本发明涉及根据上述权利要求所述的一种文件管理方法，其特征在于，数据文件分布在 Q 个存储单元中，每个存储单元对应于 P 层目录，每个目录包含 N 个目录。

与专利申请 US2004/0236761 的教导相反，该方案导致缩减了树形图的各层的目录的数量，这允许改进性能（尤其是在二维树形图中存取数据文件的速度）并使得可被记录的数据文件的数量能够几乎不受限制。该限制不再取决于文件管理系统的限制，而仅受存储装置的物理容量限制。

与现有技术的方案相反，本发明通过应用哈希函数，不再在给定的目录中分配地址，而是在具有多个层的树形图中分配路径。在最后一层的子目录中的位置不再由哈希函数分配，而是由与数据文件的识别码相关联的名称分配。

优选地，N 等于 16，哈希函数是 SHA1 函数。

文件管理的一个补充问题涉及在两个应用同时存取同一数据文件的情况下保护数据的完整性。

在大多数操作系统中，已知在文件系统层次上锁定数据文件，从而允许在写入或读取过程中暂时禁止存取数据文件。这些方案由文件系统通常基于驻留在存储器中的锁定表来管理。这些方案的缺点在于，在通过网络存取文件的情况下，从多个远程计算机存取同一文件时，可能产生干扰或错误。

专利 US6850969 提出允许避免使用锁定的一种特别方案。该专利提出经验性监控一个应用在打开的数据文件中记录修改的可能性，并保证在称为原子操作的所述操作期间，没有任何其他应用介入同一数据文件。在本专利的意义上，原子操作是指以连续方式执行的不可分割的任务集，在称为原子操作的全部任务结束之前，没有被中断和受第三方操作干扰的可能。

该方案令人不满意，因为当多个应用同时对公共文件集进行处理时，该方案阻止某些操作完成。尽管该方案允许避免改变现有的数据文件，但会有损于并发的安全的存取。实际上，该方案要求每个处理都在上一处理完成之后介入。因此该方案确实不适于对包含例如 XML 格式的数据的文件进行处理，所述文件可能在任何时间被不同的并发应用所使用。

为此，本发明还涉及一种文件管理方法，其特征在于，每个数据文件包括头部和主体。头部包括数据文件在文件管理系统中的状态参数。主体包括文件的可修改内容。该方法包括存取目标数据文件的步骤，使所述状态参数更改为禁止新的存取的状态。

对所述数据文件的管理独立于计算机操作系统的文件系统。

通过阅读下面与非限定性实施例相对应的描述，并参照附图，可更好地理解本发明，在附图中：

- 图 1 示出根据本发明的文件管理方法实施的树形图的示意图，

- 图 2 至图 8 示出文件管理的功能示意图，

- 图 9 示出与客户-服务器模式相对应的、通过网络使用文件管理系统的示意图。

图 1 示出根据本发明的文件管理方法实施的树形图的示意图。

在所描述的示例中，按照细分为 M = 5 层（1 至 5）的结构来组织数据文件。除最后一层 5 之外，每层（1 至 4）的每个目录都包含 N = 16 个目录（10 至 12）、（20 至 25）、（30 至 35）、（40 至 45）。16 个目录中每一个的名称对应于 0 和 f 之间的十六进制数字。结构中包含的目录的总数为 $16 + 16^2 + 16^3 + \ldots\ldots + 16^M$。

在所描述的示例中，树形图包括：

- 层 1 到 4 的 69904 个目录，不包含数据文件，而仅包含上层的目录；

- 层 5 的 1048576 个目录（50 至 55）；每个目录包含允许存储数据文件的文件。

最后一层 5 的各目录（50 至 55）可以包含数量 L 受限的数据文件，L 被选为较小，例如约为 1000，以便允许快速的存取时间，无论何种计算机操作系统的文件系统都可适用。

为数据文件分配存储器位置

当创建由字符串形成的独特使用名称所表示的新数据文件时，本方法包括对代表了使用名称的该字符串应用哈希函数 H_i，该步骤的结果由以十六进制形式表示的值构成。对于哈希函数 SHA1，十六进制形式的结果包括 40 个 0 和 f 之间的十六进制数字。

本方法包括，在前面限定的目录的树形图的情况下，基于来自哈希函数的所述结果来在所述树形图中确定目录的路径。

来自哈希函数的结果的起始 M 个数字的每个数字的值（rang）对应于目录树形图的层次。

如果以使用名称“myfile”为例，则应用 SHA1 函数返回值“b3580ab45cb088ba47ff070aa81c2dae1be56ca2”。

层 1 的目录是带有名称“b”的目录，对应于 SHA1 值的第一个字符，层 2 的目录是“3”，对应于第二个字符，对于 5 个层（1 至 5）以此类推。在最后一层目录 5 中的位置由构成使用名称的各个字符的八位字节的具有两个数字的十六进制表示构成。对于名称“myfile”，文件的位置和名称是“6d7966696c65”，“6d”对应于字符“m”的具有两个数字的十六进制表示，“79”对应于字符“y”的具有两个数字的十六进制表示，“66”对应于字符“f”的具有两个数字的十六进制表示，以此类推。

对于需要用多个八位字节来编码的字符，八位字节的整体将依次以这种形式表示。

由于数据文件的使用名称的独特性，文件的表示必然是独特的，这是因为该表示产生于基于哈希函数计算出的位置与以独特的方式基于使用名称计算出的数据文件名称的组合。

该方案允许以等概率的方式并且以最大离散度将数据文件分配到目录树形图的层 M 的目录中。只有层 M 包含数据文件，中间层不包含数据文件并仅用于将文件分配到层 M 的目录中。

在层 M 的各目录内部，数据文件的数量是适当的，例如约为 1000，这导致了快速的存取时间。可以在层 M 的树形图中创建的数据文件的最大数量等于 1000×16^M。例如，对于 M = 5，可以在层 M 的目录树形图中创建、写入或读取约十亿个数据文件。

数据文件的创建、写入、读取和删除

第一步骤包括，通过应用前面提到的哈希函数以及基于使用名称计算

文件名称，来基于文件的使用名称确定文件在树形图中的存取路径及其相关的文件名称。该步骤允许确定文件被记录在层 M 的哪一目录中及文件在该目录中的位置。

对同一数据文件的并发存取

为了允许安全地管理对数据文件的存取，数据文件中每一个都包括头部和主体。主体包括文件的可以以直接存取的方式或以压缩或加密的方式修改的内容。该主体前面置有头部，该头部包含文件在文件管理系统中的状态参数。

系统性地，为了进行一系列任意操作（写入、读取、修改或删除），由处理对文件的存取表现为第一获取步骤，该第一获取步骤包括由另一个处理将头部的所述状态参数修改为禁止新的存取的状态。

当然，当文件处于由一个处理进行存取或使用过程中时，另一个处理在执行该初始获取步骤时将遇到禁止或等待提示。

另一个处理反复执行该初始获取步骤，直到目标文件重新处于允许存取的状态，或直到预定的期限引起存取尝试的中断。

一旦由处理完成了一系列操作，最后一个释放步骤将头部中的经修改的参数恢复为初始状态，从而允许另一个处理对数据文件进行存取。

图 2 至图 8 示出由本发明实施的功能结构。

获取现有数据文件的功能

图 2 示出用于获取现有数据文件所需的任务序列。

该功能采用如上所述基于使用名称而计算出的文件名称作为输入 150。

第一任务 100 包括尝试在操作系统的文件系统层次上同时打开和锁定文件，通过要求打开和锁定文件阻止后面的处理打开和锁定该文件，直到关闭和解锁该文件。

该功能随后进行测试 101 以验证文件的打开和锁定：如果文件未被打开和锁定，则测试 101 的结果是否定的并且该功能进行第二测试 102，第二测试 102 验证在进行尝试 100 时文件是否存在。在否定的回答的情况下，该功能送回错误 103，错误 103 指示文件不存在。

在肯定的回答的情况下，测试 104 验证尝试 100 的次数是否超过阈值，

或是否超过了时间期限，在否定的回答的情况下，在延时 149 之后进行新的尝试 100；否则，该功能发回系统错误 105。

如果测试 101 的结果是肯定的，则该功能在任务 106 中在文件头部的状态参数中读取文件的可用性状态。

该功能随后根据状态参数的值来进行文件可用性的测试 107。

如果测试 107 的结果是否定的，这对应于不可用状态，则该功能执行在操作系统的文件系统层次上同时关闭和解锁文件的任务 108。

该功能随后进行测试 109，测试 109 验证尝试 100 的次数是否超过阈值，或是否超过了时间期限，在否定的回答的情况下，在延时 149 之后进行新的尝试 100；否则，该功能发回错误 110，错误 110 指示文件仍由当前处理占用。

如果测试 107 的结果是肯定的，则该功能执行将文件的不可用状态写入文件的头部的状态参数中的任务 111，然后，执行计算文件获取识别码的任务 112，将该识别码写入文件的头部的状态参数中的任务 113，以及读取文件主体的任务 114。任务 115 对应于对文件的主体的处理，以便提取文件的可修改内容。该处理根据文件的头部的状态参数来执行。该处理例如对应于对文件的可修改内容的解压或解密。最后的任务 116 包括在操作系统的文件系统层次上同时关闭和解锁文件。

该功能在结束 151 时作为输出送回对数据文件获取的确认、文件获取识别码以及文件的可修改内容。

获取新的数据文件的功能

图 3 示出用于获取新的数据文件所需的任务序列。

该功能采用如上所述基于使用名称而计算出的文件名称作为输入 250。

第一任务 200 包括尝试在操作系统的文件系统层次上同时创建、打开和锁定新的文件，通过要求创建、打开和锁定文件阻止后面的处理打开和锁定该文件，直到关闭和解锁该文件。

该功能随后进行测试 201 以验证文件的创建、打开和锁定：如果文件未被创建、打开和锁定，则测试 201 的结果是否定的并且该功能进行第二测试 202，第二测试 202 验证在尝试 200 时文件是否存在。

在肯定的回答的情况下，该功能发回错误 204，指示文件已经存在。

在否定的回答的情况下，该功能发回指示系统错误的错误 203。

如果测试 201 的结果是肯定的，则该功能执行创建和写入具有空文件主体的文件的头部的任务 205，然后执行在文件的头部的状态参数中写入文件的不可用状态的任务 206，然后执行计算文件获取识别码的任务 207，将该识别码写入文件的头部的状态参数中的任务 208。最后的任务 209 包括在操作系统的文件系统层次上同时关闭和解锁文件。

该功能在结束 251 时，作为输出送回对数据文件获取的确认和文件获取识别码。

获取现有数据文件（如果所述数据文件不存在则创建数据文件）的功能

图 4 示出用于获取现有数据文件（如果不存在则创建数据文件）所需的任务序列。

该功能采用如上所述基于使用名称而计算出的文件名称作为输入 350。

第一任务 300 包括尝试在操作系统的文件系统层次上同时打开和锁定文件，如果文件不存在则创建文件，通过要求打开和锁定文件阻止后面的处理打开和锁定该文件，直到关闭和解锁该文件。

该功能随后进行测试 301 以验证文件的打开和锁定：如果文件未被打开和锁定，则测试 301 的结果是否定的并且该功能进行第二测试 302，第二测试 302 验证在尝试 300 时文件是否存在。在否定的回答的情况下，该功能发回系统错误 303。

在肯定的回答的情况下，测试 304 验证尝试 300 的次数是否超过阈值，或是否超过了时间期限，在否定的回答的情况下，在延时 349 之后进行新的尝试 300；否则，该功能发回系统错误 305。

如果测试 301 的结果是肯定的，则该功能随后进行测试 306 以验证在尝试 300 中是否创建了文件：如果未创造文件，则测试 306 的结果是否定的并且该功能在执行任务 307 时在文件的头部的状态参数中读取文件的可用性状态。

该功能随后根据状态参数的值来进行文件可用性的测试 308。

如果测试 308 的结果是否定的，这对应于不可用状态，则该功能执行在操作系统的文件系统层次上同时关闭和解锁文件的任务 309。

该功能随后进行测试310，测试310验证尝试300的次数是否超过阈值，或是否超过了时间期限，在否定的回答的情况下，在延时349之后进行新的尝试300；否则，该功能发回错误311，错误311指示文件仍由当前处理占用。

如果测试308的结果是肯定的，则该功能执行将文件的不可用状态写入文件的头部的状态参数中的任务312，然后执行计算文件获取识别码的任务313，将该识别码写入文件的头部的状态参数中的任务314，以及读取文件主体的任务315。任务316对应于对文件主体的处理，以便提取文件的可修改内容。该处理根据文件的头部的状态参数来进行。该处理例如对应于对文件的可修改内容的解压或解密。任务317包括在操作系统的文件系统层次上同时关闭和解锁文件。

该功能在结束351时，作为输出送回对数据文件获取的确认、表示未创建文件的通知、文件获取识别码以及文件的可修改内容。

如果测试306的结果是肯定的，则该功能执行创建和写入具有空文件主体的文件的头部的任务318，然后执行将文件的不可用状态写入文件的头部的状态参数中的任务319，然后执行计算文件获取识别码的任务320，将该识别码写入文件的头部的状态参数中的任务321。最后的任务322包括在操作系统的文件系统层次上同时关闭和解锁文件。

该功能在结束352时作为输出送回对数据文件获取的确认、表示创建了文件的通知以及文件获取的识别码。

在不修改数据的情况下释放获取的数据文件的功能

图 5 示出用于在不改变数据的情况下释放获取的数据文件所需的任务序列。

该功能采用如上所述基于使用名称而计算出的文件名称以及在前述获取数据文件时送回的获取识别码作为输入450。

第一任务 400 包括尝试在操作系统的文件系统层次上同时打开和锁定文件，通过要求打开和锁定文件阻止后面的处理打开和锁定该文件，直到关闭和解锁该文件。

该功能随后进行测试401以验证文件的打开和锁定：如果文件未被打开和锁定，则测试401的结果是否定的并且该功能进行第二测试402，第二测试402验证在尝试400时文件是否存在。在否定的回答的情况下，该功能送回错误403，错误403指示文件不存在。

在肯定的回答的情况下，测试404验证尝试400的次数是否超过阈值，或是否超过了时间期限，在否定的回答的情况下，在延时449之后进行新的尝试400；否则，该功能送回系统错误405。

如果测试401的结果是肯定的，则该功能在执行任务406时在文件的头部的状态参数中读取文件的可用性状态。

该功能随后根据状态参数的值来进行文件可用性的测试407。

如果测试407的结果是肯定的，则该功能执行在操作系统的文件系统层次上同时关闭和解锁文件的任务408，并送回错误409，错误409表示未获取文件。

如果测试407的结果是否定的，这对应于不可用状态，则该功能在执行任务410时，读取文件的头部中的获取识别码。

该功能随后进行测试411，测试411检查该识别码是否不同于作为输入提交的识别码。在肯定的回答的情况下，该功能执行在操作系统的文件系统层次上同时关闭和解锁文件的任务412，并且送回错误413，错误413表示作为输入提交的获取识别码无效。

在否定的回答的情况下，该功能执行删除文件的头部的状态参数中的获取识别码的任务414，然后执行将可用性状态写入文件的头部中的任务415。最后的任务416包括在操作系统的文件系统层次上同时关闭和解锁文件。

该功能在结束451时作为输出送回对数据文件释放的确认。

在修改数据的情况下释放获取的数据文件的功能

图 6 示出用于在修改数据的情况下释放获取的数据文件所需的任务序列。

该功能采用如上所述基于使用名称而计算出的文件名称、在前述获取数据文件时送回的获取识别码以及文件的可修改内容作为输入550。

第一任务 500 包括尝试在操作系统的文件系统层次上同时打开和锁定文件，通过要求打开和锁定文件阻止后面的处理打开和锁定该文件，直到关闭和解锁该文件。

该功能随后进行测试501以验证文件的打开和锁定：如果文件未被打开和锁定，则测试501的结果是否定的并且该功能进行第二测试502，第二测试502验证在尝试500时文件是否存在。在否定的回答的情况下，该

功能送回错误 503，错误 503 指示文件不存在。

在肯定的回答的情况下，测试504验证尝试500的次数是否超过阈值，或是否超过了时间期限，在否定的回答的情况下，在延时 549 之后进行新的尝试 500；否则，该功能送回系统错误 505。

如果测试 501 的结果是肯定的，则该功能在执行任务 506 时在文件的头部的状态参数中读取文件的可用性状态。

该功能随后根据状态参数的值来进行文件可用性的测试 507。

如果测试 507 的结果是肯定的，则该功能执行在操作系统的文件系统层次上同时关闭和解锁文件的任务 508，并且送回错误 509，错误 509 指示未获取文件。

如果测试 507 的结果是否定的，这对应于不可用状态，则该功能在执行任务 510 时，读取文件的头部中的获取识别码。

该功能随后进行测试 511，测试 511 检查该识别码是否不同于作为输入提交的识别码。在肯定的回答的情况下，该功能执行在操作系统的文件系统层次上同时关闭和解锁文件的任务 512，并且送回错误 513，错误 513 表示作为输入提交的获取识别码无效。

在否定的回答的情况下，该功能执行删除文件的头部的状态参数中的获取识别码的任务 514，然后执行将可用性状态写入文件的头部中的任务 515，然后执行与对文件主体的处理相对应的任务 516，以便将文件的可修改内容插入到文件主体中。该处理是根据文件的头部的状态参数来进行的并且对应于例如对文件的可修改内容的压缩或加密。该功能随后进行写入文件主体的任务 517，最后的任务 518 包括在操作系统的文件系统层次上同时关闭和解锁文件。

该功能在结束 551 时作为输出送回对数据文件的释放的确认以及对文件内容的修改的确认。

在删除获取的数据文件的情况下释放所述数据文件的功能

图 7 示出用于在删除获取的数据文件的情况下释放所述数据文件所需的任务序列。

该功能采用如上所述基于使用名称而计算出的文件名称以及在前述获取数据文件时送回的获取识别码作为输入 650。

第一任务 600 包括尝试在操作系统的文件系统层次上同时打开和锁

定文件，通过要求打开和锁定文件阻止后面的处理打开和锁定该文件，直到关闭和解锁该文件。

该功能随后进行测试 601 以验证文件的打开和锁定：如果文件未被打开和锁定，则测试 601 的结果是否定的并且该功能进行第二测试 602，第二测试 602 验证在尝试 600 时文件是否存在。在否定的回答的情况下，该功能送回错误 603，错误 603 指示文件不存在。

在肯定的回答的情况下，测试 604 验证尝试 600 的次数是否超过阈值，或是否超过了时间期限，在否定的回答的情况下，在延时 649 之后进行新的尝试 600；否则，该功能送回系统错误 605。

如果测试 601 的结果是肯定的，则该功能在执行任务 606 时在文件的头部的状态参数中读取文件的可用性状态。

该功能随后根据状态参数的值来进行文件可用性的测试 607。

如果测试 607 的结果是肯定的，则该功能执行在操作系统的文件系统层次上同时关闭和解锁文件的任务 608，并且送回错误 609，错误 609 指示未获取文件。

如果测试 607 的结果是否定的，这对应于不可用状态，则该功能在执行任务 610 时读取文件的头部中的获取识别码。

该功能随后进行测试 611，测试 611 检查该识别码是否不同于作为输入提交的识别码。在肯定的回答的情况下，该功能执行在操作系统的文件系统层次上同时关闭和解锁文件的任务 612，并且送回错误 613，错误 613 表示在输入处提交的获取识别码无效。

在否定的回答的情况下，该功能执行删除文件的头部的状态参数中的获取识别码的任务 614，然后执行将可用性状态写入文件的头部中的任务 615。最后的任务 616 包括在操作系统的文件系统层次上同时关闭、解锁和删除文件。

该功能在结束 651 时作为输出送回对数据文件释放的确认以及对数据文件删除的确认。

在不获取现有数据文件的情况下简单读取现有数据文件的功能

图 8 示出用于在不获取现有数据文件的情况下简单读取现有数据文件所需的任务序列。

该功能采用如上所述基于使用名称而计算出的文件名称作为输入

750。

第一任务 700 包括尝试在操作系统的文件系统层次上同时打开和锁定文件，通过要求打开和锁定文件阻止后面的处理打开和锁定该文件，直到关闭和解锁该文件。

该功能随后进行测试 701 以验证文件的打开和锁定：如果文件未被打开和锁定，则测试 701 的结果是否定的并且该功能进行第二测试 702，第二测试 702 验证在尝试 700 时文件是否存在。在否定的回答的情况下，该功能送回错误 703，错误 703 指示文件不存在。

在肯定的回答的情况下，测试 704 验证尝试 700 的次数是否超过阈值，或是否超过了时间期限，在否定的回答的情况下，在延时 749 之后进行新的尝试 700；否则，该功能送回系统错误 705。

如果测试 701 的结果是肯定的，则该功能在执行任务 706 时在文件的头部的状态参数中读取文件的可用性状态。

该功能随后根据状态参数的值来进行文件可用性的测试 707。

如果测试 707 的结果是否定的，这对应于不可用状态，则该功能执行在操作系统的文件系统层次上同时关闭和解锁文件的任务 708。

该功能随后进行测试 709，测试 709 验证尝试 700 的次数是否超过阈值，或是否超过了时间期限，在否定的回答的情况下，在延时 749 之后进行新的尝试 700；否则，该功能送回错误 710，错误 710 指示文件仍由当前处理占用。

如果测试 707 的结果是肯定的，则该功能执行读取文件主体的任务 711，随后执行处理文件主体的任务 712，以便提取文件的可修改内容。该处理是根据文件的头部的状态参数来进行的。该处理对应于例如对文件的可修改内容的解压或解密。最后的任务 713 包括在操作系统的文件系统层次上同时关闭和解锁文件。

该功能在结束 751 时作为输出送回对数据文件的简单读取的确认以及文件的可修改内容。

其它的功能允许便于管理树形图，尤其是：

－读取数据文件的头部的状态参数，包括文件的可用性状态，

－强制释放未由处理释放的数据文件，

－通过树形图的子集列出树形图中包含的数据文件。

客户服务器模式

图 9 涉及一种特别的和优选的应用场合。图 9 示出用于通过网络使用根据本发明的文件管理系统（该模式对应于客户服务器模式）所需的任务序列。

在该场合中，文件管理系统安装在包括服务器软件 801 的服务器计算机 800 上。服务器计算机可由多个客户计算机 802 至 804 访问，所述客户计算机每个都包括客户软件 805 至 807。客户软件和服务器软件之间的交互使用已知类型的网络传输协议，例如 HTTP 或优选为 HTTPS 类型的安全协议。

在该场合中，本发明使用应用协议 808，应用协议 808 允许通过“文件获取”或“文件释放”类型的指令来调用符合本发明的文件管理系统的功能。文件以多个子集 809 至 811 的形式分布在服务器计算机上，每个子集对应于根据本发明的称为“表”的树形图。数据文件的使用名称在协议中表示为“键”。

通过客户端软件在客户计算机中的一台上执行第一任务 812。第一任务 812 包括调用针对表和键的获取、释放或简单读取的功能。

通过服务器软件在服务器计算机上执行第二任务 813。第二任务 813 包括执行由客户软件所请求的功能。

通过服务器软件在服务器计算机上执行第三任务 814。第三任务包括向客户软件发回所请求的功能的结果。

可选地，系统包括多个服务器，从而允许分配数据的存储负荷和容量。在该场合中，客户软件 805 至 807 包括根据表和键来选择服务器的规则。

服务器软件 801 还可以包括对执行过的操作的日志功能，以便允许在备份服务器上增量重构服务器计算机的树形图，而无需在备份服务器上对树形图进行完整的复制。

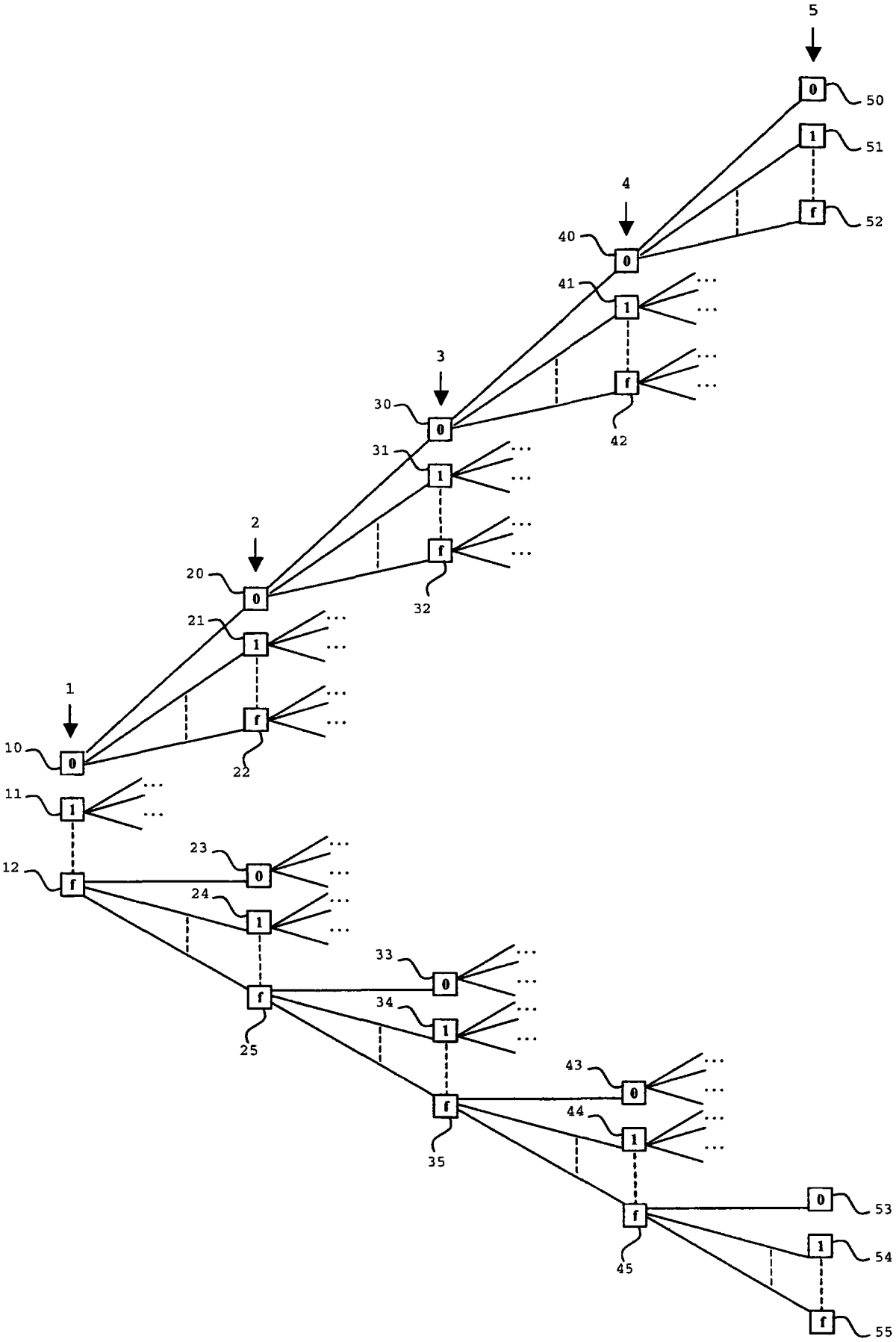

1
2
3
4
5
10
11
12
20
21
22
23
24
25
30
31
32
33
34
35
40
41
42
43
44
45
50
51
52
53
54
55
0
1
f

图1

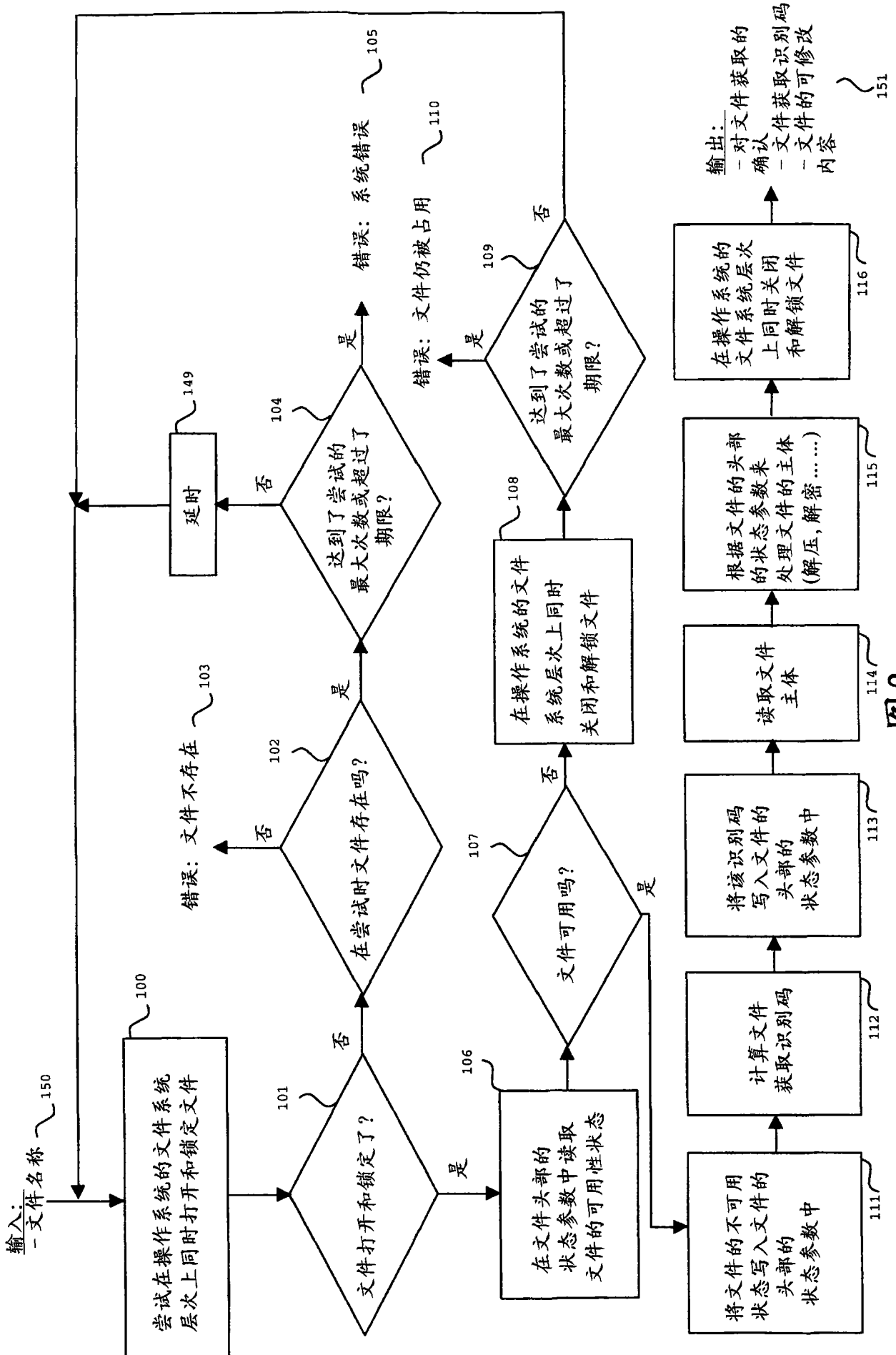

输入:
- 文件名称
150
尝试在操作系统的文件系统层次上同时打开和锁定文件
100
文件打开和锁定了?
101
否
是
在尝试时文件存在吗?
102
否
是
错误: 文件不存在
103
达到了尝试的最大次数或超过了期限?
104
否
是
延时
149
错误: 系统错误
105
在文件头部的状态参数中读取文件的可用性状态
106
文件可用吗?
107
否
是
在操作系统的文件系统层次上同时关闭和解锁文件
108
达到了尝试的最大次数或超过了期限?
109
否
是
错误: 文件仍被占用
110
将文件的不可用状态写入文件的头部的状态参数中
111
计算文件获取识别码
112
将该识别码写入文件的头部的状态参数中
113
读取文件主体
114
根据文件的头部的状态参数来处理文件的主体(解压,解密……)
115
在操作系统的文件系统层次上同时关闭和解锁文件
116
输出:
- 对文件获取的确认
- 文件获取识别码
- 文件的可修改内容
151

图2

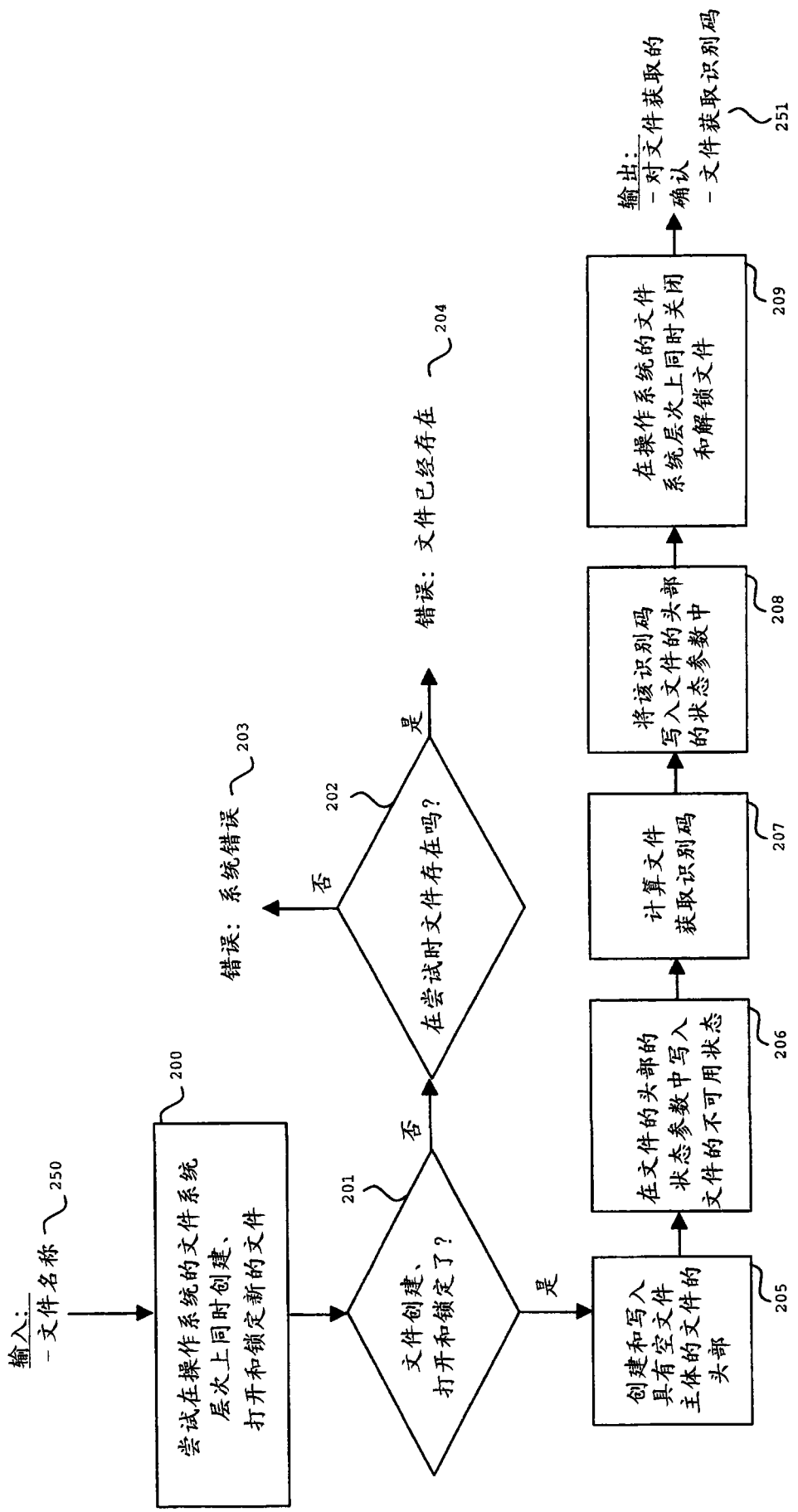

输入:
- 文件名称
250
尝试在操作系统的文件系统层次上同时创建、打开和锁定新的文件
200
文件创建、打开和锁定了?
201
否
是
错误: 系统错误
203
在尝试时文件存在吗?
202
否
是
错误: 文件已经存在
204
创建和写入具有空文件主体的文件的头部
205
在文件的头部的状态参数中写入文件的不可用状态
206
计算文件获取识别码
207
将该识别码写入文件的头部的状态参数中
208
在操作系统的文件系统层次上同时关闭和解锁文件
209
输出:
- 对文件获取的确认
- 文件获取识别码
251

图 3

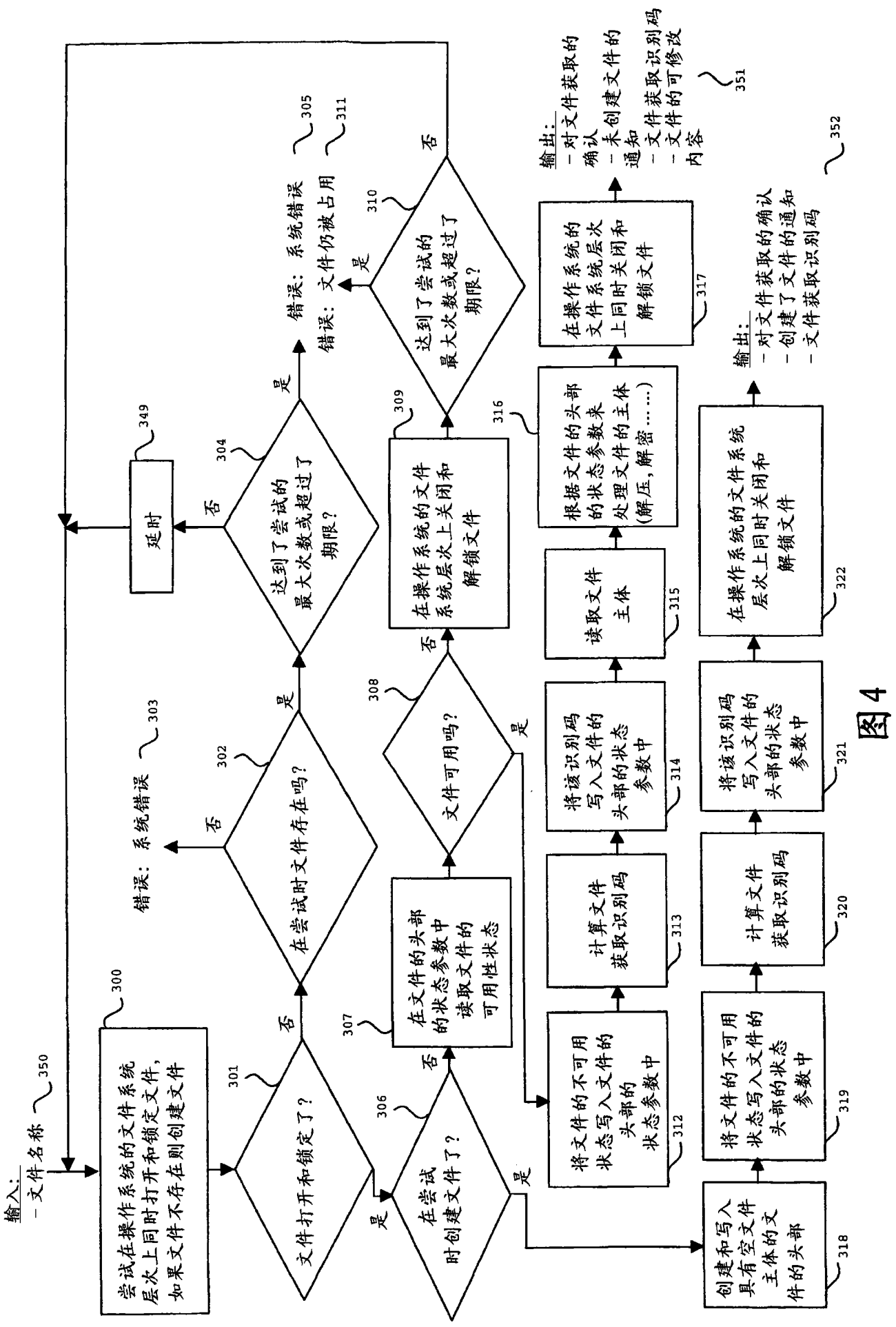

输入:
- 文件名称
350
尝试在操作系统的文件系统层次上同时打开和锁定文件，如果文件不存在则创建文件
300
文件打开和锁定了？
301
否
是
在尝试时文件存在吗？
302
否
错误：系统错误
303
是
达到了尝试的最大次数或超过了期限？
304
否
延时
349
是
错误：系统错误
305
错误：文件仍被占用
311
在尝试时创建文件了？
306
否
是
在文件的头部的状态参数中读取文件的可用性状态
307
文件可用吗？
308
否
是
达到了尝试的最大次数或超过了期限？
310
否
是
在操作系统的文件系统层次上同时打开和锁定文件，如果文件不存在则创建文件
309
创建和写入具有空文件主体的文件的头部
318
将文件的不可用状态写入文件的头部的状态参数中
319
将文件的不可用状态写入文件的头部的状态参数中
312
计算文件获取识别码
313
计算文件获取识别码
320
将该识别码写入文件的头部的状态参数中
314
将该识别码写入文件的头部的状态参数中
321
根据文件的头部的状态参数来处理文件的主体(解压，解密……)
316
读取文件主体
315
在操作系统的文件系统层次上同时关闭和解锁文件
322
在操作系统的文件系统层次上同时关闭和解锁文件
317
输出:
- 对文件获取的确认
- 创建了文件的通知
- 文件获取识别码
352
输出:
- 对文件获取的确认
- 未创建文件的通知
- 文件获取识别码
- 文件的可修改内容
351

图 4

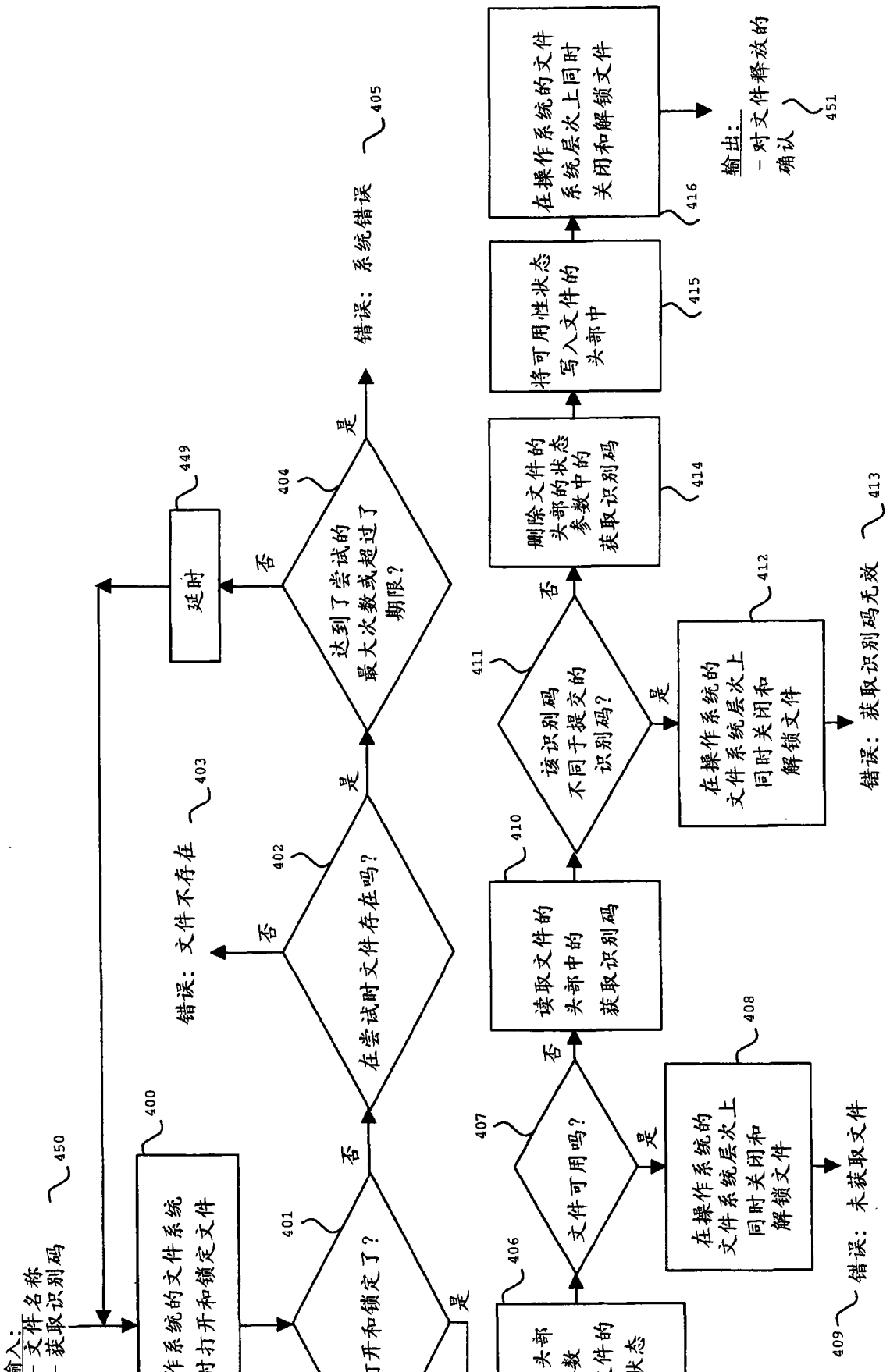

输入:
- 文件名称
- 获取识别码
450
尝试在操作系统的文件系统层次上同时打开和锁定文件
400
文件打开和锁定了?
401
否
是
在尝试时文件存在吗?
402
否
是
错误: 文件不存在
403
达到了尝试的最大次数或超过了期限?
404
否
是
错误: 系统错误
405
延时
449
在文件的头部的状态参数中读取文件的可用性状态
406
文件可用吗?
407
否
是
读取文件的头部中的获取识别码
410
该识别码不同于提交的识别码?
411
否
是
错误: 获取识别码无效
413
删除文件的头部的状态参数中的获取识别码
414
将可用性状态写入文件的头部中
415
在操作系统的文件系统层次上同时关闭和解锁文件
416
输出:
- 对文件释放的确认
451
在操作系统的文件系统层次上同时关闭和解锁文件
412
错误: 未获取文件
409
在操作系统层次上同时关闭和解锁文件
408

图5

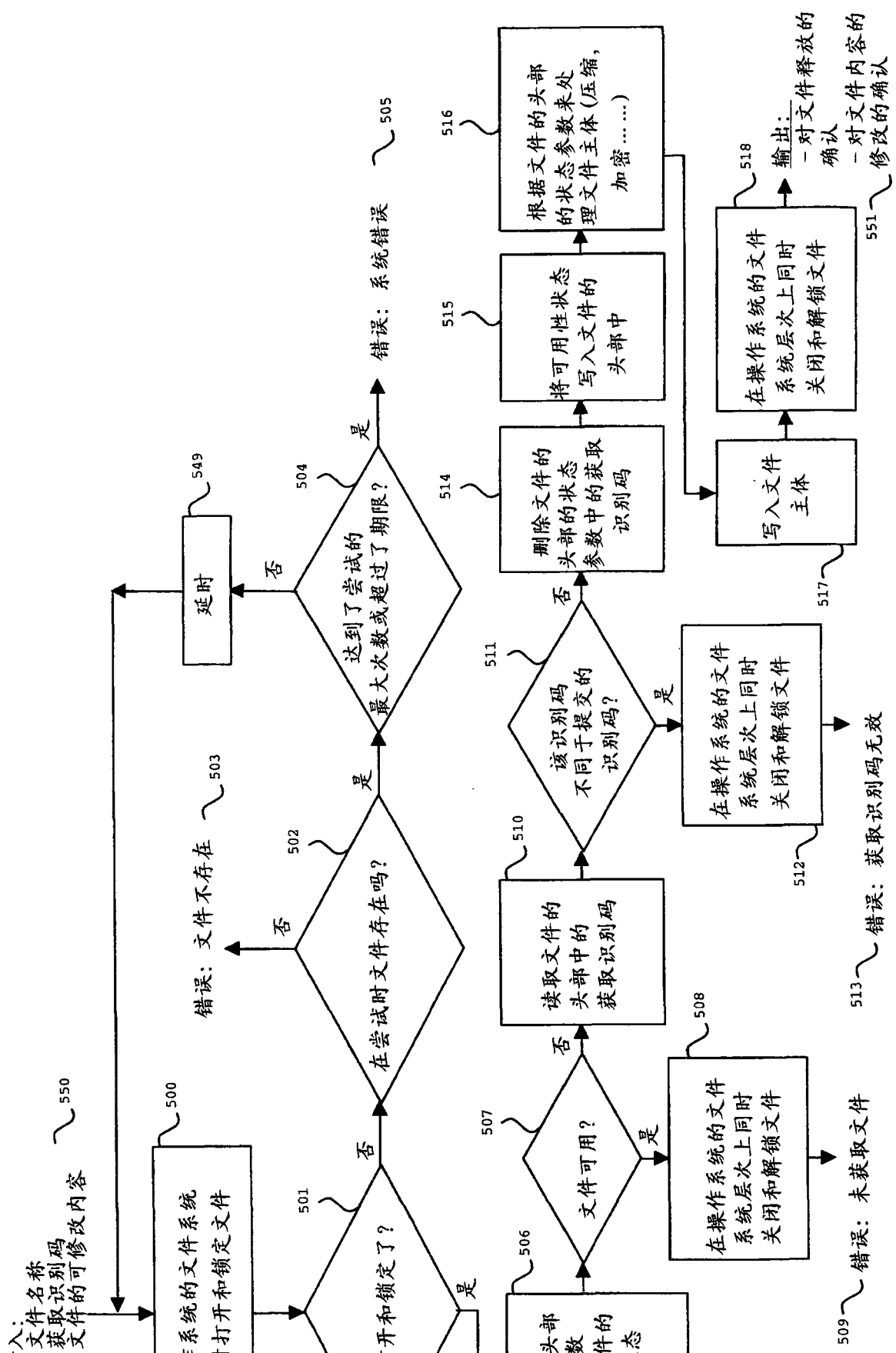

输入：
- 文件名称
- 获取识别码
- 文件的可修改内容
550
尝试在操作系统的文件系统层次上同时打开和锁定文件
500
文件打开和锁定了？
501
否
是
在尝试时文件存在吗？
502
否
是
错误：文件不存在
503
达到了尝试的最大次数或超过了期限？
504
否
是
延时
549
错误：系统错误
505
在文件的头部的状态参数中读取文件的可用性状态
506
文件可用？
507
否
是
读取文件的头部中的获取识别码
510
该识别码不同于提交的识别码？
511
否
是
删除文件的头部的状态参数中的获取识别码
514
将可用性状态写入文件的头部中
515
根据文件的头部的状态参数来处理文件主体（压缩，加密……）
516
写入文件主体
517
在操作系统层次上同时关闭和解锁文件
518
输出：
- 对文件释放的确认
- 对文件内容的修改的确认
551
在操作系统层次上同时关闭和解锁文件
508
在操作系统的文件系统层次上同时关闭和解锁文件
512
错误：未获取文件
509
错误：获取识别码无效
513

图6

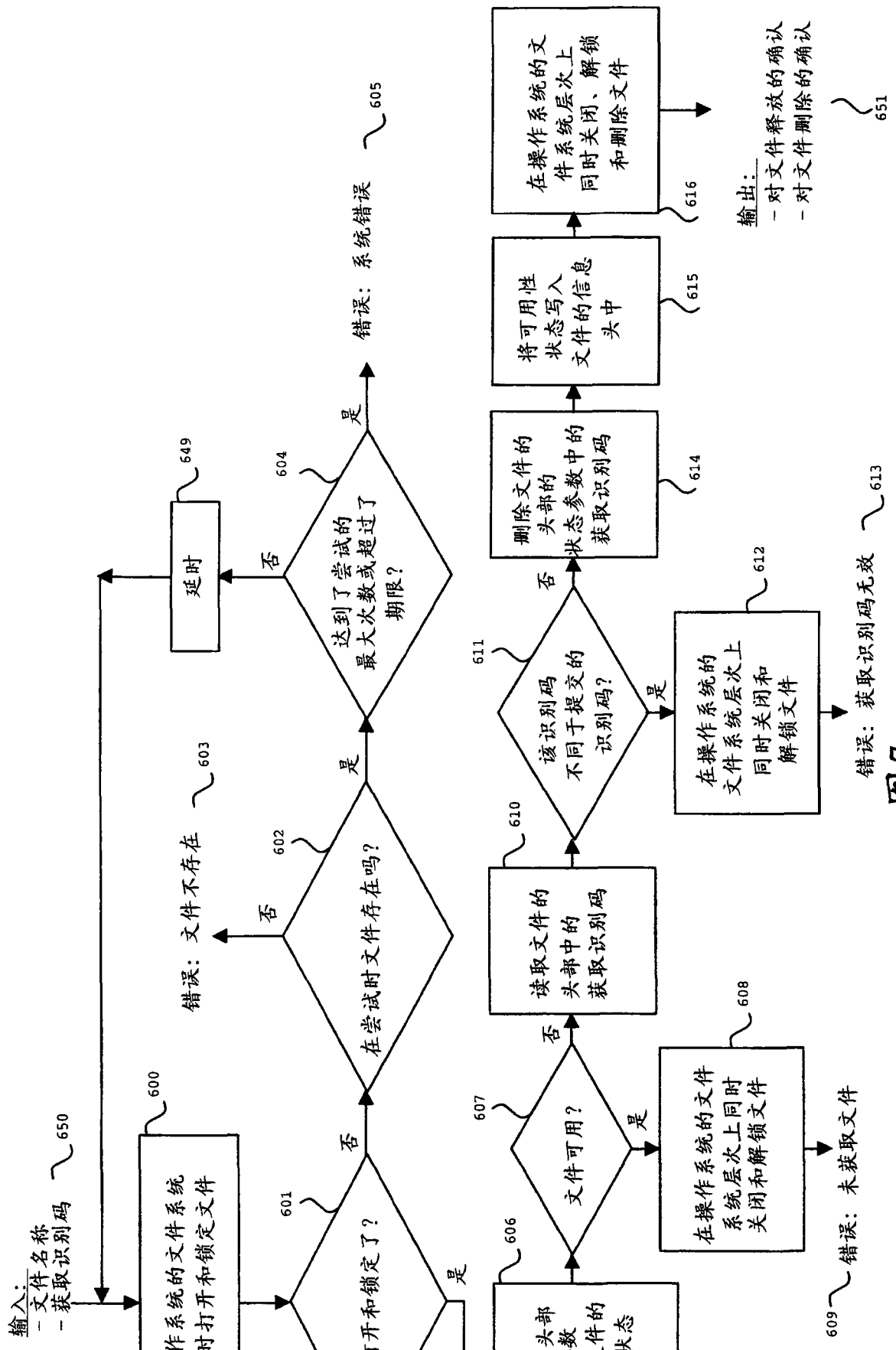

输入：
- 文件名称
- 获取识别码
650
尝试在操作系统的文件系统层次上同时打开和锁定文件
600
文件打开和锁定了？
601
否
是
在文件的头部的状态参数中读取文件的可用性状态
606
文件可用？
607
否
是
读取文件的头部中的获取识别码
610
该识别码不同于提交的识别码？
611
否
是
删除文件的头部的状态参数中的获取识别码
614
将可用性状态写入文件的信息头中
615
在操作系统的文件系统层次上同时关闭、解锁和删除文件
616
输出：
- 对文件释放的确认
- 对文件删除的确认
651
在操作系统的文件系统层次上同时关闭和解锁文件
608
错误：未获取文件
609
在操作系统的文件系统层次上同时关闭和解锁文件
612
错误：获取识别码无效
613
在尝试时文件存在吗？
602
否
是
错误：文件不存在
603
达到了尝试的最大次数或超过了期限？
604
否
是
延时
649
错误：系统错误
605

图7

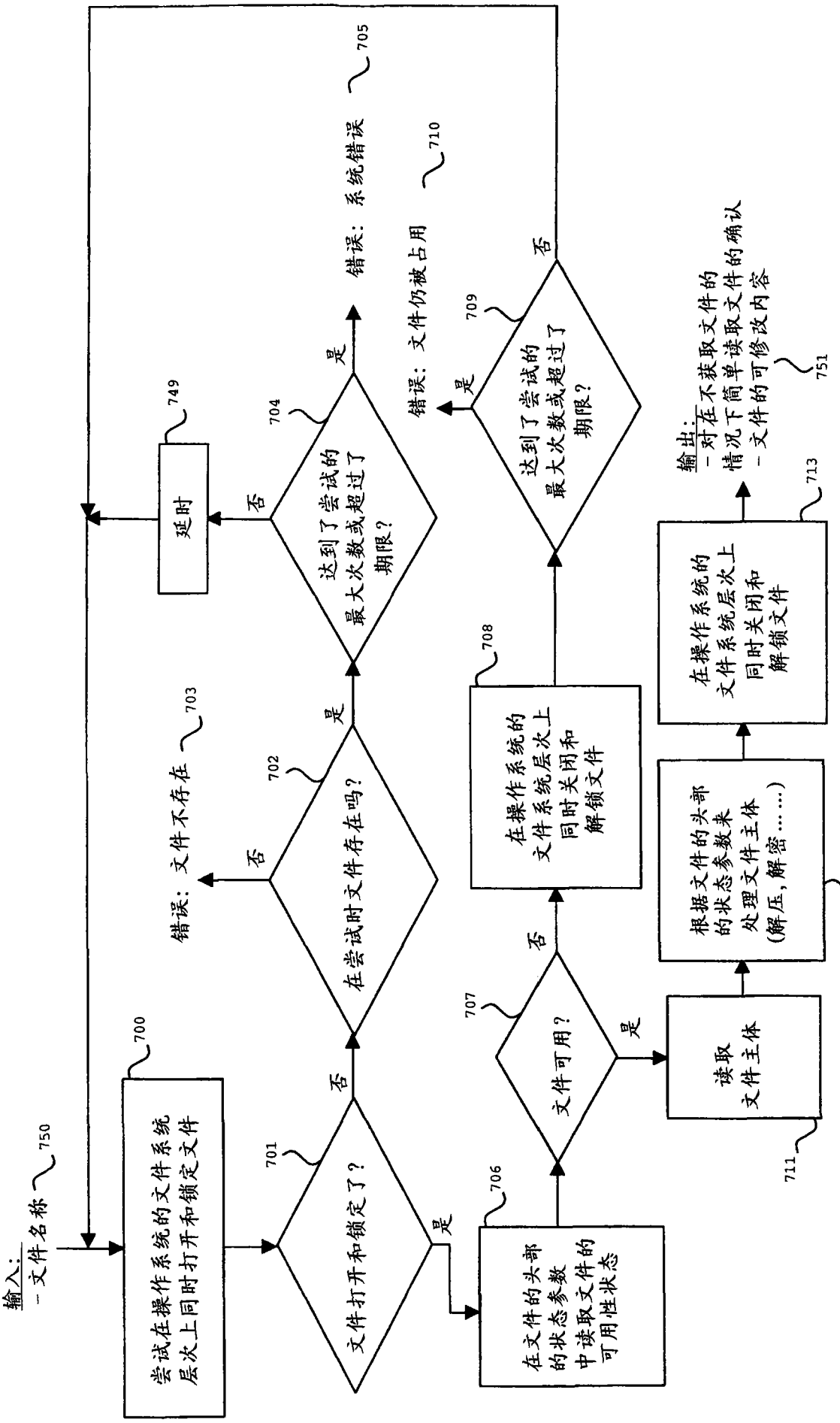

输入:
- 文件名称
750
尝试在操作系统的文件系统层次上同时打开和锁定文件
700
文件打开和锁定了?
701
否
是
在尝试时文件存在吗?
702
否
是
错误: 文件不存在
703
达到了尝试的最大次数或超过了期限?
704
否
是
延时
749
错误: 系统错误
705
在文件的头部的状态参数中读取文件的可用性状态
706
文件可用?
707
否
是
在操作系统的文件系统层次上同时关闭和解锁文件
708
达到了尝试的最大次数或超过了期限?
709
否
是
错误: 文件仍被占用
710
读取文件主体
711
根据文件的头部的状态参数来处理文件主体(解压,解密.....)
712
在操作系统的文件系统层次上同时关闭和解锁文件
713
输出:
- 对在不获取文件的情况下简单读取文件的确认
- 文件的可修改内容
751

图8

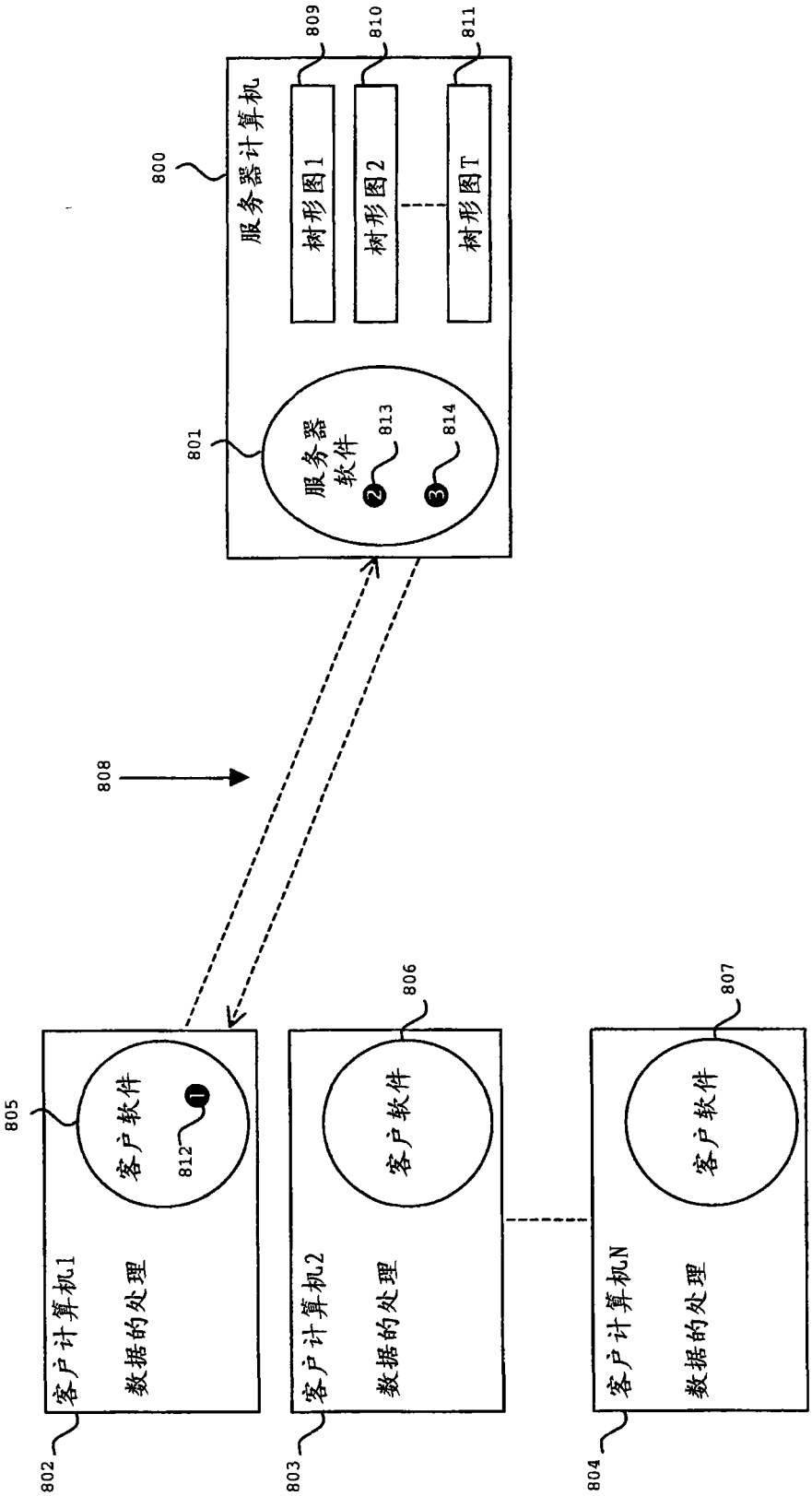

800
服务器计算机
809
810
811
树形图1
树形图2
树形图T
801
服务器软件
813
814
808
805
客户软件
812
客户计算机1
数据的处理
802
806
客户软件
客户计算机2
数据的处理
803
807
客户软件
客户计算机N
数据的处理
804

图9