



[12] 发明专利说明书

专利号 ZL 200710149183.3

[45] 授权公告日 2009 年 12 月 9 日

[11] 授权公告号 CN 100568186C

[22] 申请日 2007.9.5

[21] 申请号 200710149183.3

[30] 优先权

[32] 2006.9.8 [33] US [31] 11/530,087

[73] 专利权人 国际商业机器公司

地址 美国纽约

[72] 发明人 E·S·本德尔

[56] 参考文献

US 5745676A 1998.4.28

CN 1808325A 2006.7.26

US 5974550A 1999.10.26

US 2003/0188174A1 2003.10.2

US 2004/0268317A1 2004.12.30

审查员 沈乐平

[74] 专利代理机构 北京市中咨律师事务所

代理人 于 静 李 峰

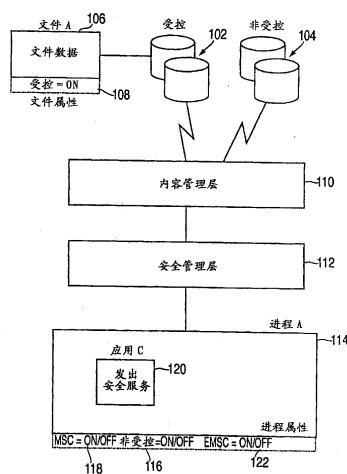
权利要求书 3 页 说明书 8 页 附图 3 页

[54] 发明名称

用于实现进程间完整性串行化的方法和系统

[57] 摘要

一种用于实现进程间完整性串行化服务的方法、系统和计算机程序产品被提供。所述方法包括：当确定仅被指定为受控的程序(如果存在的话)已被加载用于调用进程时，对于所述调用进程使能包括必须保持受控(MSC)状态和扩展必须保持受控(EMSC)状态的进程状态。所述调用进程请求将目标程序加载到临时存储中用于实施安全服务。基于所述目标程序的控制指示符、MSC 状态和 EMSC 状态，所述方法包括控制所述临时存储中一个或更多活动。所述活动包括：将所述目标程序加载到所述临时存储中，执行所述临时存储中的主程序，以及在所述调用进程的生命周期期间贯穿该主程序的执行而重置所述 MSC 状态和 EMSC 状态。



1. 一种用于实现进程间完整性串行化服务的方法，包括：

当确定仅被指定为受控的程序已被加载用于调用进程时，对于所述调用进程使能包括必须保持受控状态和扩展必须保持受控状态的进程状态，所述调用进程请求将目标程序加载到临时存储中用于实施安全服务；以及

基于所述目标程序的控制指示符、必须保持受控状态和扩展必须保持受控状态，控制所述临时存储中的一个或更多活动，其包括：

将所述目标程序加载到所述临时存储中，其中所述临时存储对于所述调用进程可访问；

执行所述临时存储中的主程序；以及

在所述调用进程的生命周期期间贯穿所述主程序的执行而重置所述必须保持受控状态和扩展必须保持受控状态；

其中，基于按照安全准则的验证，程序被指定为受控的。

2. 根据权利要求 1 所述的方法，其中，所述主程序在控制被传递给另一程序之前清除所述调用进程的临时存储。

3. 根据权利要求 1 所述的方法，其中，当所述控制指示符反映出所述目标程序是受控的并且所述调用进程的必须保持受控状态被使能时，所述目标程序被加载到所述临时存储中。

4. 根据权利要求 1 所述的方法，进一步包括：

当所述目标程序的控制指示符反映出该目标程序是非受控的以及当所述必须保持受控状态被禁用时，设置所述调用进程的进程控制属性以反映出非受控；以及

将所述目标程序加载到所述临时存储中；

其中，如果未按照安全准则被验证或响应于按照所述安全准则的验证失败，则程序被指定为非受控的。

5. 根据权利要求 1 所述的方法，进一步包括：

一旦接收到对执行所述主程序的请求，则：

当确定所述扩展必须保持受控状态被禁用时，重置该调用进程的所述必须保持受控状态；以及

授权所述主程序的执行。

6. 根据权利要求 1 所述的方法，进一步包括：

一旦接收到对执行所述主程序的请求，则：

当确定所述扩展必须保持受控状态被使能以及所述目标程序的控制属性反映出该目标程序是非受控的时，拒绝对执行该主程序的请求。

7. 根据权利要求 1 所述的方法，进一步包括：

一旦接收到对执行所述目标程序的请求，则：

当确定所述扩展必须保持受控状态被使能以及所述目标程序的控制属性反映出该目标程序是受控的时，保存该调用进程的所述必须保持受控状态和扩展必须保持受控状态。

8. 一种用于实现进程间完整性串行化服务的系统，包括：

配置为当确定仅被指定为受控的程序已被加载用于调用进程时，对于所述调用进程使能包括必须保持受控状态和扩展必须保持受控状态的进程状态的装置，所述调用进程请求将目标程序加载到所述计算机处理系统的临时存储中用于实施安全服务；以及

配置为基于所述目标程序的控制指示符、必须保持受控状态和扩展必须保持受控状态，控制所述临时存储中一个或更多活动的装置，其包括：

配置为将所述目标程序加载到所述临时存储中，其中所述临时存储对于所述调用进程可访问的装置；

配置为执行所述临时存储中的主程序的装置；以及

配置为在所述调用进程的生命周期期间贯穿所述主程序的执行而重置所述必须保持受控状态和扩展必须保持受控状态的装置；

其中，基于按照安全准则的验证，程序被指定为受控的。

9. 根据权利要求 8 所述的系统，进一步包括配置为使得所述主程序在控制被传递给另一个程序之前清除所述调用进程的所述临时存储的装置。

10. 根据权利要求 8 所述的系统，进一步包括配置为使得当所述控制

指示符反映出所述目标程序是受控的并且所述调用进程的必须保持受控状态被使能时，所述目标程序被加载到所述临时存储中的装置。

11. 根据权利要求 8 所述的系统，进一步包括：

配置为当所述目标程序的控制指示符反映出该目标程序是非受控的并且所述必须保持受控状态被禁用时，设置所述调用进程的进程控制属性以反映出非受控的装置；以及

配置为将所述目标程序加载到所述临时存储中的装置；

其中，如果未按照安全准则被验证或响应于按照所述安全方针的验证失败，则程序被指定为非受控的。

12. 根据权利要求 8 所述的系统，进一步包括：

配置为一旦接收到对执行所述主程序的请求，则：

当确定所述扩展必须保持受控状态被禁用时，重置所述调用进程的必须保持受控状态；以及

授权所述目标程序的执行的装置。

13. 根据权利要求 8 所述的系统，其中，进一步包括：

配置为一旦接收到对执行所述主程序的请求，则：

当所述扩展必须保持受控状态被使能并且所述目标程序的控制属性反映出该目标程序为是非受控的时，拒绝所述对执行该主程序的请求的装置；以及

配置为一旦接收到对执行所述主程序的请求，则：

当确定所述扩展必须保持受控状态被使能并且所述目标程序的控制属性反映出该目标程序为受控的时，保存该调用进程的所述必须保持受控状态和扩展必须保持受控状态的装置。

用于实现进程间完整性串行化的方法和系统

技术领域

本公开一般涉及地址空间通信，并且特别涉及用于为建立进程间的信任水平而实现进程间完整性串行化的方法、系统和计算机程序产品。

背景技术

有时需要跨多个进程的应用来交换敏感数据和/或提供可以被相互请求的一组已授权命令。当应用中的进程之一已错误地从非受保护库加载程序时，问题出现。一旦该应用中的一个进程失去完整性，则该整个应用可能不再是安全的，由此使敏感数据有风险。所述的例子如登录应用的守护程序（daemon）进程（例如父进程），其中，该进程创建将代表客户端进行动作的子进程。所述子进程执行已授权的客户端代码，所述代码然后将开始与其父进程（例如守护程序进程）的信息交换。在事件的过程中，所述子进程不经意地从非受控的库加载可执行程序，并且被感染。所述守护程序父进程继续接受来自该受感染子进程的请求，由此导致可能的安全破坏。

因此，所需要的是这样一种方法，该方法用于在一个或更多进程要求敏感数据的交换时建立持续所述进程的生命周期的进程间的信任。

发明内容

本发明的实施例包括用于实现进程间完整性串行化服务的方法。所述方法包括：当确定仅被指定为受控的程序（如果存在的话）已被加载用于调用进程时，对于所述调用进程使能包括必须保持受控（MSC）状态和扩展必须保持受控（EMSC）状态的进程状态。所述调用进程请求将目标程

序加载到临时存储中用于实施安全服务。基于所述目标程序的控制指示符、MSC 状态和 EMSC 状态，所述方法包括控制所述临时存储中一个或更多活动。所述活动包括：将所述目标程序加载到所述临时存储中，执行所述临时存储中的主程序，以及在所述调用进程的生命周期期间贯穿该主程序的执行而重置所述 MSC 状态和 EMSC 状态。

通过审阅下面的附图和详细描述，根据实施例的其它系统、方法和/或计算机程序产品对于本领域的技术人员将变得显而易见。应当明白，所有所述另外的系统、方法和/或计算机程序产品被包括在本说明书中、在本发明的范围内、并且受权利要求保护。

附图说明

被认作本发明的主题在本说明书的结论处的权利要求中被特别指出和清楚要求保护。根据下面详细描述并参考附图，本发明的前述和其它目的、特征和优点将是显而易见的，在附图中：

图 1 是根据示例性实施例的系统的一部分的框图，其中，进程间完整性串行化服务可以在所述系统上实现；

图 2 是描述在示例性实施例中用于经由进程间完整性串行化服务设置持久受控状态的过程的流程图；以及

图 3 是示出了在示例性实施例中用于经由进程间完整性串行化服务使能两个进程之间的敏感或受保护信息的交换的过程的流程图。

详细描述借助于例子并参考附图阐明了本发明的优选实施例以及优点和特征。

具体实施方式

进程间完整性串行化服务根据示例性实施例被提供。所述进程间完整性串行化服务提供了一种方法，在该方法中，进程可以明确设置地址空间（临时存储）中的状态，其防止从任何未授权库进行加载。该状态在这里称为扩展必须保持受控（MSC）状态。一旦该状态被设置，则所述进程可

以创建子进程，由此该扩展 MSC (EMSC) 状态被传播到该子进程。该 EMSC 被配置使得其不能被任何程序装置禁用，由此确保所述两个进程（例如所述父进程和子进程）的生命周期中不可以出现未授权加载。也由所述进程间完整性串行化服务提供接口，以允许所述子进程查询该新状态，以及如果新状态被启用，则向所述子进程确保其父进程也是安全的。

现在转向图 1，现在将描述根据示例性实施例的系统的一部分，其中，所述进程间完整性串行化服务可以在该系统上实现。图 1 的系统可以是计算机处理系统（例如大型计算机或其它高性能多处理器设备）的一部分，以及包括：数据库 102、104，内容管理层 110，安全管理层 112 以及相互通信的一个或更多进程（例如进程 A 114）。所述进程间完整性串行化服务可以经由在所述计算机处理系统上执行的一个或更多应用被实现为独立的软件工具。可选地，所述进程间完整性串行化服务可以经由内容管理层 110、安全管理层 112 或其组合、经由适用于在这些系统单元中使用的程序代码来实现。

所述进程在可由内容管理层 110 和安全管理层 112 寻址的临时存储单元中被实现。图 1 的系统允许安装经由外部属性将卷、数据集或单独文件标记为“受控”的能力，其中，所述外部属性可以由安装定义的管理用户设置。

经由图 1 的系统实现的进程间完整性串行化服务防止了进程（例如进程 A 114）当处于 MSC 状态时加载非受控程序或文件。受控文件被定义为已被系统程序员验证为满足该文件被执行所在的计算机网络的所有安全/完整性准则的文件（例如没有病毒/蠕虫）。而非受控程序可以是不可用于一般系统使用的新开发程序（例如针对个人使用编写的新工具）。其实，受控程序可以认为是可信的，而非受控程序不是。具有管理权限的系统用户可以负责确定该系统的数据存储库（例如数据库 102 和 104）中的哪些文件或程序（例如文件 106）是受控的。受控文件（例如文件 A 106）使用外部属性设置 108（例如“受控 = ON”）来标记。该外部属性设置这里称为“控制指示符”。

所述系统的內容管理層 110 負責確保：當請求進程已被標記為 MSC（例如 MSC 使能）時，僅“受控”程序被加載到該請求進程（例如進程 A 114）中。進程可以被定義為一個或更多操作，所述操作為產生所希望的結果的目的而利用某種資源。內容管理層 110 可以經由用於管理內容的捕獲、存儲、檢索等的中間件應用來實現。內容可以包括應用、文件、文檔等。

如圖 1 的系統中所示，進程 114 包括指定該進程的當前狀態的進程屬性：MSC 屬性 118、非受控屬性 116 和 EMSC 屬性 122。內容管理層 110 還負責由於“非受控”程序的成功加載而標記該進程為“非受控”（經由屬性 116）。在常規的 MSC 使能的系統中，進程的所述“非受控”狀態只可以通過來自受控或非受控庫（例如數據庫 102/104）的主程序（例如應用 C 120）的執行來重置。主程序可以被定義為這樣的一種程序，其中，在控制被給予新的可執行程序之前，調用進程（例如進程 A 114）的所有應用存儲被清除。與所述常規的 MSC 使能的系統不同，所述進程間完整性串行化服務如這裡所描述的那樣在所述進程的生命周期中防止了“非受控”狀態的重置。

內容管理層 110 與所述系統的安全管理層 112 協商以確定文件（例如文件 A 106）是否可以被加載到進程（例如進程 A 114）中。安全管理層 112 可以使用用於保護所述系統的數據和應用的一組功能來實現。安全管理層 112 負責使得“非受控”進程（例如當非受控屬性 116 = ON 時）中的應用（例如應用 C 120）所嘗試的、被認為是“安全”相關的任何服務失敗。安全管理層 112 還負責：當任何“安全”相關服務執行時將進程標記為 MSC（經由 MSC 屬性 118）。如所述“非受控”狀態一樣，常規的 MSC 使能的系統僅當主程序執行時提供對所述 MSC 狀態的重置。然而，所述進程間完整性串行化服務在進程的生命周期中防止了所述進程受控狀態的重置。通過防止該重置，所述進程將在該進程的生命周期中保持受控。

如上面指出的，所述進程間完整性串行化服務使進程能夠明確地設置用於防止從任何未授權庫的加載的 EMSC 狀態。所述 EMSC 狀態是貫穿

主程序的执行并且在所述进程的生命周期中持续的永久受控状态，如现在将关于图 2 描述的那样。如图 2 的流程图中所示，所述进程间完整性串行化服务经由安全管理层 112 结合调用进程（例如进程 A）和内容管理层 110 来实施。

所述进程间完整性串行化提供这样一种服务，该服务被创建并且允许对所述 EMSC 状态的明确设置（经由属性 122）。该服务不需要任何特别授权，因为其目要目的是对所述调用进程（例如进程 A）施加另外的文件访问限制。在步骤 202，所述调用进程（例如进程 A 114）请求所述 EMSC 状态的使能。一旦被调用，则安全管理层 112 在步骤 204 检查该调用进程是否之前已加载或执行了来自非受控库（例如数据库 104）的程序。如果是这样，则该服务将失败（即 EMSC 使能请求被拒绝），其通知调用者（例如进程 A）该进程已是受控的。在所述情况下，该进程执行进行到步骤 210。

然而，如果所述调用进程在步骤 204 还未加载或执行来自非受控库的程序，则该调用进程在步骤 208 被分别经由属性 118 和 122 标记为 MSC 和 EMSC。所述 MSC 设置 118 维护其历史功能（例如按照常规的 MSC 使能的系统），以及所述 EMSC 设置 122 添加如这里描述的那样防止非受控主程序的执行被执行的新功能。除对主程序执行的所添加的限制外，所述 EMSC 状态将导致该 EMSC 状态和 MSC 状态 118、122 都贯穿主程序的执行被传播。如上面指出的，在常规系统中，主程序的执行将重置所述进程受控状态。所述进程间完整性串行化服务防止所述进程受控状态的重置。通过防止该重置，所述进程将在该进程的生命周期中保持受控。

在步骤 210，所述进程（例如进程 A 114）请求程序（例如应用 C 120）被加载。内容管理层 110 在步骤 212 确定目标程序（即所述将被加载的程序）是否例如经由控制指示符 108 被设置为非受控。如果是这样，则内容管理层 110 在步骤 214 确定所述进程状态是否被设置为 MSC=ON（经由属性 118）。如果不是这样，则加载所述进程的请求在步骤 216 失败，并且该进程以常规方式进行。否则，该进程在步骤 218 被标记为非受控（经

由属性 116），以及控制在步骤 213 被返回给该进程。

返回步骤 212，如果内容管理层 110 确定目标程序不是非受控的（例如来自数据库 102），则该程序在步骤 213 被加载到临时存储中，由此所述进程然后可以在需要时在不经过内容管理层 110 的情况下按照该程序进行动作。在步骤 220，所述进程请求主程序的执行（例如经由加载和执行主程序的执行系统服务和清除该进程中的所有存储）。内容管理层 110 在步骤 222 经由属性 122 确定所述进程状态是否被设置为 EMSC。如果不是，则 MSC 状态在步骤 224 被重置（即，MSC 属性 118 被设置为 OFF），并且该执行请求被授权。否则，内容管理层 110 在步骤 226 确定目标程序是否是非受控的（例如来自数据库 104）。如果是这样，则为执行该程序的请求在步骤 228 失败。否则，所述 MSC 和 EMSC 状态在步骤 230（经由属性 118、122）由内容管理层 110 保存用于该进程的剩余部分。

如上面指出的，一旦所述 EMSC 状态经由属性 122 被设置，则调用进程然后可以创建子进程，由此该 EMSC 状态被传播到该子进程。现在转向图 3，现在将根据示例性实施例描述用于经由 EMSC 状态设置 122 和进程间完整性串行化服务来使能两个进程（例如父和子进程）间的敏感信息的交换的过程。

如上面指出的，所述 EMSC 状态可以用于保护多进程应用。初始进程（例如进程 A 114）明确地使能 MSC 状态（经由属性 118），其还设置另外的 EMSC 指示符（经由属性 122）。该初始进程在步骤 302 创建子进程。所述 EMSC 和 MSC 状态（经由属性 118、122）在步骤 304 被传播到所述初始进程（父进程）创建的子进程。

在步骤 306，所述子进程如由其父进程指示的那样执行主程序代码（例如应用 C 120），而该父进程等待。一种方法由所述进程间完整性串行化服务提供给进程（例如子进程），用以查询其 EMSC 状态，从而确定其是否已从其父进程继承了所述 EMSC 状态。在步骤 308，该子进程查询其状态。如果该子进程的查询状态不是 EMSC，则在步骤 310 与其父进程的数据交换失败。如果该子进程的查询状态被指示为 EMSC，则该子进程发起

与其父进程的数据交换。在步骤 312，该父进程接收对数据交换的请求。由于该父进程在该子进程创建之前明确设置了所述 EMSC 和 MSC 状态，所以该父进程被保证：该子进程的状态继承自该父进程，以及因此可知该子进程的状态也是 EMSC 和 MSC。这时，所述两个进程在步骤 314 可以相互交换安全数据。

如上面指出的，所述 EMSC 和 MSC 状态贯穿受控主程序的执行而持续。从非受控库（例如数据库 104）加载的任何程序均不被允许经由所述进程运行，以及为完成此而作的任何尝试会导致执行失败并且终止子进程。由于所述进程间完整性串行化服务不提供经由其进程可以禁用 EMSC 状态的任何方法，所以父进程可以被确保：其子进程在该子进程的整个生命周期中是受控的。

另外，如果子进程实施所述状态查询，则其知道其当前正在运行 EMSC=ON，以及，如果该子进程自己还未明确设置所述 EMSC 状态，则该子进程可以被确保：其从其父进程继承该 EMSC 状态，并且因此可以相信其父进程是受控的。通过在所述两个进程间建立的信任水平，交换安全数据是安全的。

如上面指出的，实施例可以以计算机实现进程和用于实施所述进程的装置的形式来实施。在示例性实施例中，本发明用由一个或更多网络单元执行的计算机程序代码来实施。实施例包括包含指令的计算机程序代码，该指令被包含在例如软盘、CD-ROM、硬盘驱动器的有形介质或任何其它计算机可读存储介质中，其中，当所述计算机程序代码被加载到计算机中并由计算机执行时，该计算机变成用于实施本发明的装置。实施例包括这样的计算机程序代码，该代码例如被存储在存储介质中，被加载到计算机中和/或由计算机执行，或者通过某种传输介质被发送，例如通过电线或电缆、通过光纤或经由电磁辐射，其中，当所述计算机代码被加载到计算机中并由计算机执行时，该计算机变成用于实施本发明的装置。当被实现在通用微处理器上时，所述计算机程序代码段配置该微处理器以创建特定逻辑电路。

尽管已参考示例性实施例描述了本发明，但本领域的技术人员应当理解，在不脱离本发明的范围的情况下，可以作出各种改变，并且等价物可以替换其单元。另外，在不脱离本发明的基本范围的情况下，可以对本发明的讲授作出许多修改来适于特定情形和材料。因此，应当明白，本发明不限于作为所设想的用于实现本发明的最佳模式被公开的特定实施例，而是本发明将包括落在权利要求的范围内的所有实施例。此外，第一、第二等术语的使用不表示任何顺序或重要性，而是所述术语第一、第二等用于区分各个单元。另外，术语一个（a）、一个（an）等的使用不表示数量的限制，而是表示至少一个所引用的项目的出现。

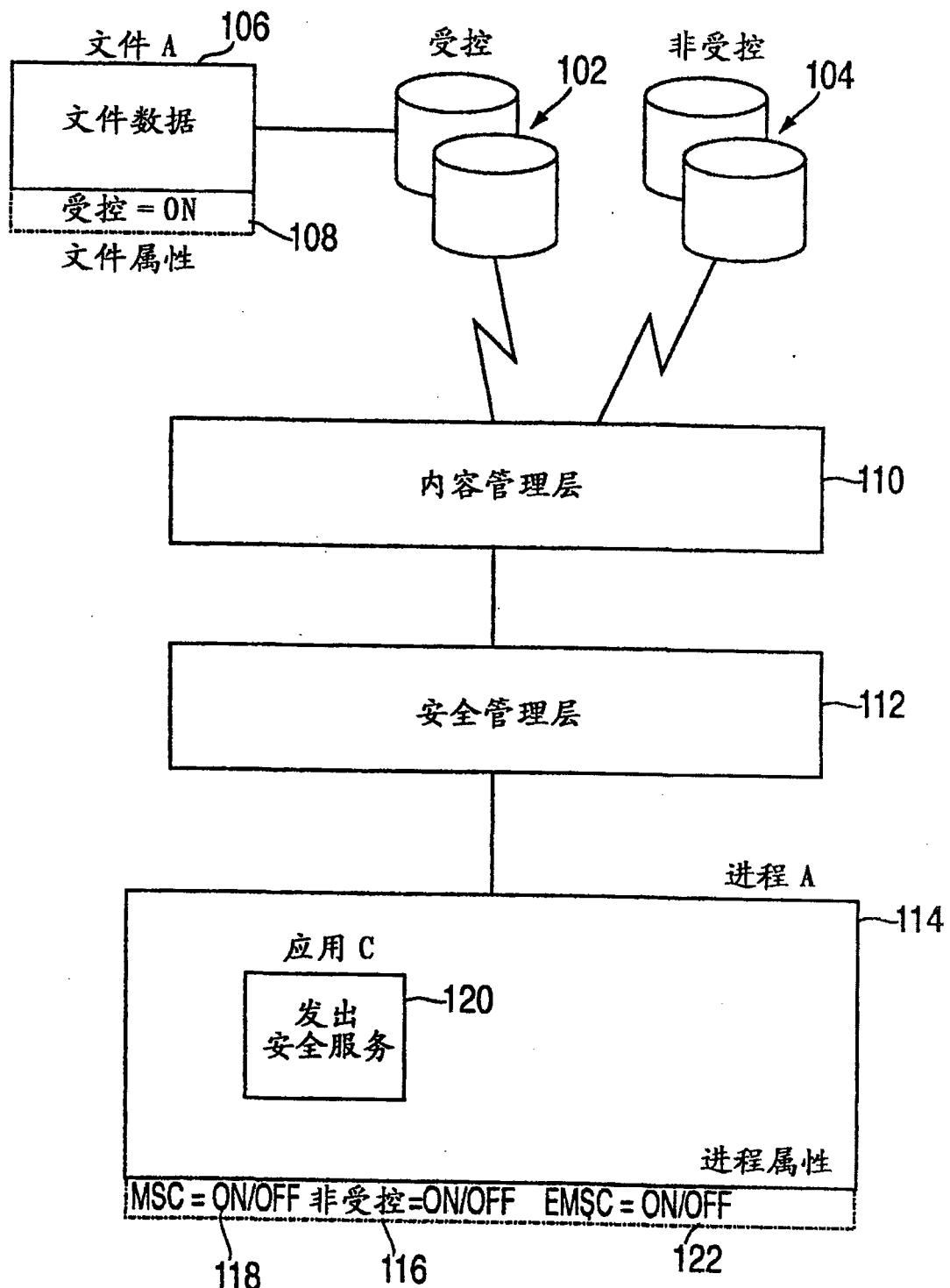


图 1

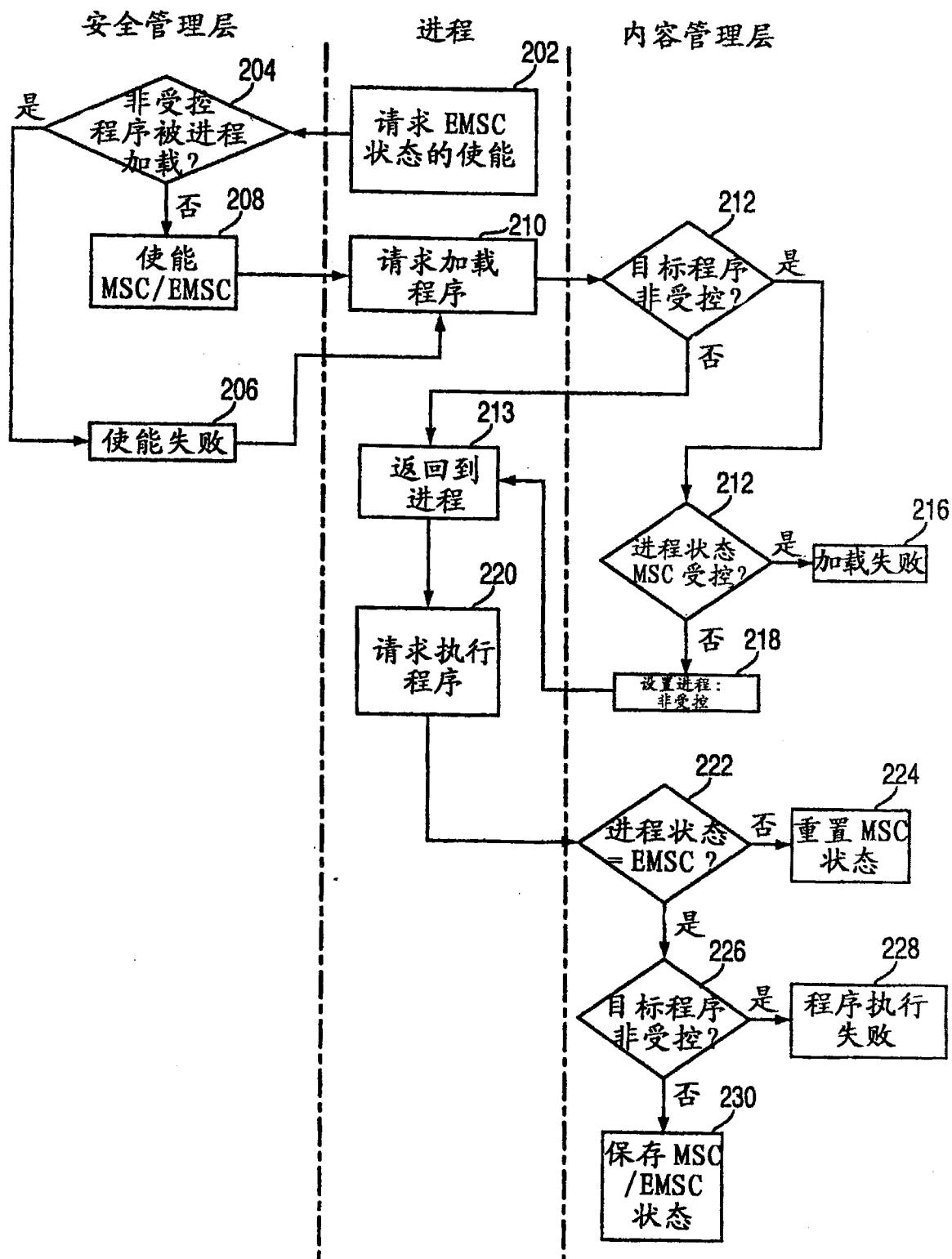


图 2

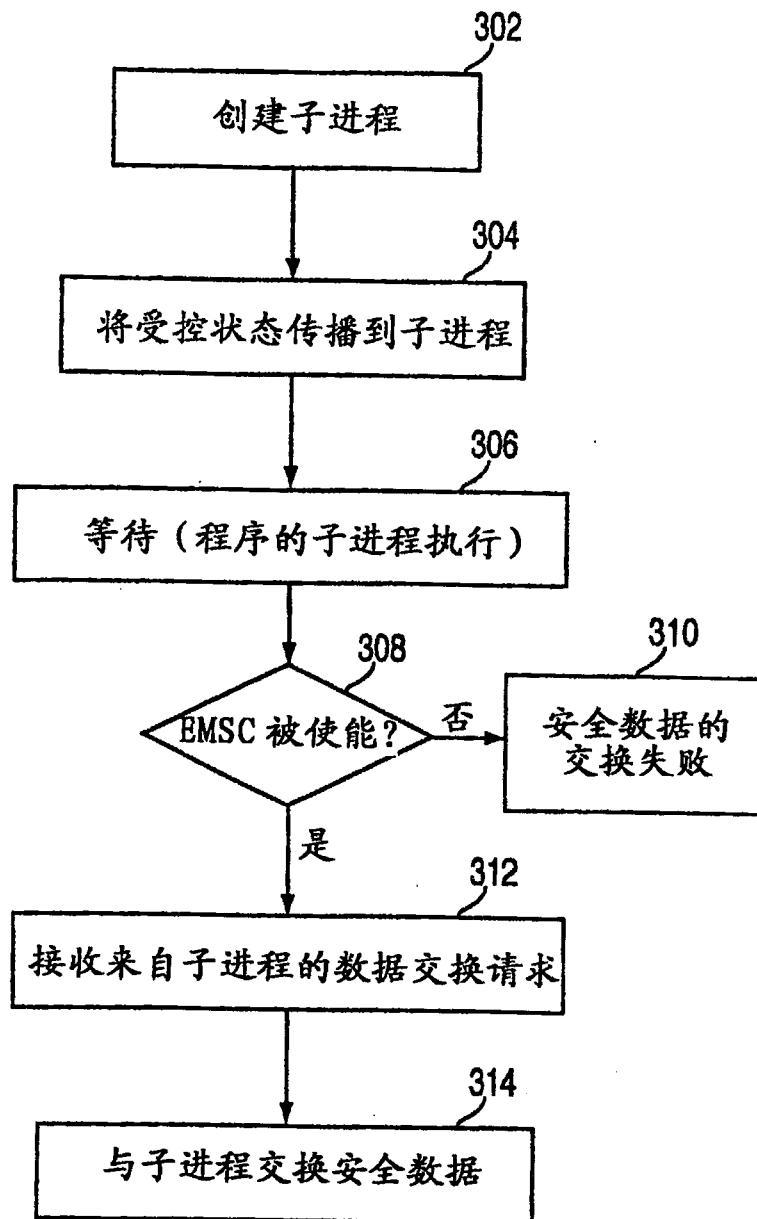


图 3