(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 104598841 A
(43)申请公布日 2015.05.06

(21)申请号 201410848638.0

(22)申请日 2014.12.29

(71)申请人 东软集团股份有限公司
地址 110179 辽宁省沈阳市浑南新区新秀街2号

(72)发明人 孟庆洋 甘凤喜

(74)专利代理机构 北京集佳知识产权代理有限公司 11227
代理人 王玲 王宝筠

(51)Int.Cl.
G06F 21/74(2013.01)
G06F 9/445(2006.01)

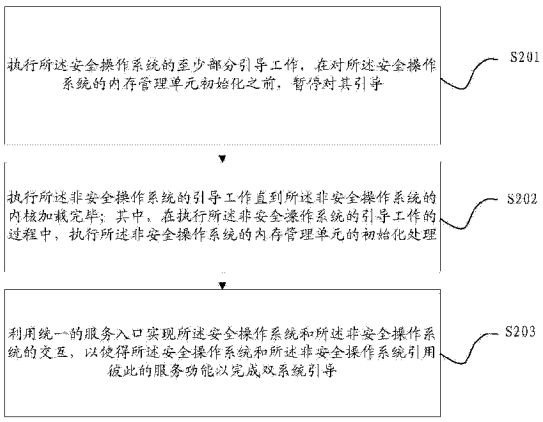
权利要求书2页 说明书8页 附图4页

(54)发明名称

一种面向终端安全的双系统引导方法和装置

(57)摘要

本发明涉及计算机技术领域，特别是一种面向终端安全的双系统引导方法和装置，所述方法包括：执行安全操作系统的至少部分引导工作，在对所述安全操作系统的内存管理单元初始化之前，暂停对其引导；执行非安全操作系统的引导工作直到所述非安全操作系统的内核加载完毕；其中，在执行所述非安全操作系统的引导工作的过程中，执行所述非安全操作系统的内存管理单元的初始化处理；利用统一的服务入口实现所述安全操作系统和所述非安全操作系统的交互，以使得所述安全操作系统和所述非安全操作系统引用彼此的服务功能以完成双系统引导。本发明可以降低安全操作系统和非安全系统的操作冲突，提高系统的可用性并减少了整个系统的引导时间，降低系统开销。



CN 104598841 A

1. 一种面向终端安全的双系统引导方法，其特征在于，所述方法应用于终端，所述终端包括安全操作系统与非安全操作系统，所述安全操作系统运行于安全执行环境中，所述非安全操作系统运行于非安全执行环境中，所述方法包括：

执行所述安全操作系统的至少部分引导工作，在对所述安全操作系统的内存管理单元初始化之前，暂停对其引导；

执行所述非安全操作系统的引导工作直到所述非安全操作系统的内核加载完毕；其中，在执行所述非安全操作系统的引导工作的过程中，执行所述非安全操作系统的内存管理单元的初始化处理；

利用统一的服务入口实现所述安全操作系统和所述非安全操作系统的交互，以使得所述安全操作系统和所述非安全操作系统引用彼此的服务功能以完成双系统引导。

2. 根据权利要求 1 所述的方法，其特征在于，所述执行所述安全操作系统的至少部分引导工作包括：

进行所述安全操作系统的待加载镜像的真实性与完整性校验；

配置所述安全操作系统的可信资源。

3. 根据权利要求 1 所述的方法，其特征在于，所述利用统一的服务入口实现所述安全操作系统和所述非安全操作系统的交互，以使得所述安全操作系统和所述非安全操作系统引用彼此的服务功能以完成双系统引导包括：

加载所述非安全操作系统的至少部分服务功能；

执行所述安全操作系统的引导工作，直到所述安全操作系统的至少部分服务功能加载完毕；其中，在执行所述安全操作系统的引导工作的过程中，所述安全操作系统通过所述统一的服务入口引用所述非安全操作系统加载的所述至少部分服务功能；

执行所述非安全操作系统的剩余引导工作，直到所述非安全操作系统加载完毕。

4. 根据权利要求 3 所述的方法，其特征在于，在执行所述非安全操作系统的剩余引导工作的过程中，所述非安全操作系统通过所述统一的服务入口引用所述安全操作系统加载的所述至少部分服务功能。

5. 根据权利要求 1 所述的方法，其特征在于，所述方法还包括：

当接收到所述非安全操作系统的调用应用指令时，根据所述调用应用指令包含的调用参数加载与所述调用参数对应的安全操作系统的对应应用。

6. 一种面向终端安全的双系统引导装置，其特征在于，所述终端包括安全操作系统与非安全操作系统，所述安全操作系统运行于安全执行环境中，所述非安全操作系统运行于非安全执行环境中，所述装置包括：

第一执行单元，用于执行所述安全操作系统的至少部分引导工作，在对所述安全操作系统的内存管理单元初始化之前，暂停对其引导；

第二执行单元，用于执行所述非安全操作系统的引导工作直到所述非安全操作系统的内核加载完毕；其中，在执行所述非安全操作系统的引导工作的过程中，执行所述非安全操作系统的内存管理单元的初始化处理；

交互引导单元，用于利用统一的服务入口实现所述安全操作系统和所述非安全操作系统的交互，以使得所述安全操作系统和所述非安全操作系统引用彼此的服务功能以完成双系统引导。

7. 根据权利要求 6 所述的装置，其特征在于，所述第一执行单元具体用于：

进行所述安全操作系统的待加载镜像的真实性与完整性校验；

配置所述安全操作系统的可信资源。

8. 根据权利要求 6 所述的装置，其特征在于，所述交互引导单元包括：

第一加载单元，用于加载所述非安全操作系统的至少部分服务功能；

第二加载单元，用于执行所述安全操作系统的引导工作，直到所述安全操作系统的至少部分服务功能加载完毕；其中，在执行所述安全操作系统的引导工作的过程中，所述安全操作系统通过所述统一的服务入口引用所述非安全操作系统加载的所述至少部分服务功能；

第三加载单元，用于执行所述非安全操作系统的剩余引导工作，直到所述非安全操作系统加载完毕。

9. 根据权利要求 8 所述的装置，其特征在于，所述第三加载单元，具体用于在执行所述非安全操作系统的剩余引导工作的过程中，通过所述统一的服务入口引用所述安全操作系统加载的所述至少部分服务功能。

10. 根据权利要求 6 所述的装置，其特征在于，所述装置还包括：

应用加载单元，用于当接收到所述非安全操作系统的调用应用指令时，根据所述调用应用指令包含的调用参数加载与所述调用参数对应的安全操作系统的对应应用。

一种面向终端安全的双系统引导方法和装置

技术领域

[0001] 本发明涉及计算机处理技术领域，特别是涉及一种面向终端安全的双系统引导方法和装置。

背景技术

[0002] 随着物联网技术的快速发展与移动设备的不断普及，智能移动终端在人们的日常生活中发挥着越来越重要的作用。随着智能移动终端的广泛应用，如何保证智能移动终端的安全性成为一个研究的热点。

[0003] 目前，大部分的移动终端采用 ARM（英文全称为 Advanced RISC Machine，中文名称为高级精简指令集机器）架构，其支持一种 ARM Trustzone（中文名称为高级精简指令集机器可信区域）技术，其将 CPU 及相关硬件资源划分为安全状态和非安全状态，在安全状态下运行安全操作系统，在非安全状态下运行非安全操作系统，例如富操作系统。在基于 ARM Trustzone 技术的终端安全解决方案中，安全操作系统与非安全操作系统的引导是一个非常关键的环节，其是建立可信链条、验证系统真实性和完整性的重要基础。现有技术的引导方式是先引导安全操作系统等到安全操作系统完全运行后，再对非安全操作系统进行校验与引导。

[0004] 发明人在实现本发明的过程中发现，现有技术存在的方法，其是先引导安全操作系统待安全操作系统完全运行后再对非安全操作系统进行引导，然而，安全操作系统在初始化过程中需要对 MMU（英文全称为 Memory Management Unit，中文全称为内存管理单元）进行一些敏感指令操作，该类操作将会导致非安全操作系统在初始化过程中对 MMU 的大量操作无效或者异常。为了保证非安全操作系统正常引导和运行，需要大范围修改非安全操作系统的内核代码，不仅使得双系统在各平台上的移植工作大量增加，还会加剧非安全操作系统内核的碎片化，增大系统开销。

发明内容

[0005] 为解决上述技术问题，本发明公开了一种面向终端安全的双系统引导方法和装置，可以降低安全操作系统和非安全系统的操作冲突，降低系统开销。

[0006] 技术方案如下：

[0007] 根据本发明实施例的第一方面，公开了一种面向终端安全的双系统引导方法，所述方法应用于终端，所述终端包括安全操作系统与非安全操作系统，所述安全操作系统运行于安全执行环境中，所述非安全操作系统运行于非安全执行环境中，所述方法包括：

[0008] 执行所述安全操作系统的至少部分引导工作，在对所述安全操作系统的内存管理单元初始化之前，暂停对其引导；

[0009] 执行所述非安全操作系统的引导工作直到所述非安全操作系统的内核加载完毕；其中，在执行所述非安全操作系统的引导工作的过程中，执行所述非安全操作系统的内存管理单元的初始化处理；

[0010] 利用统一的服务入口实现所述安全操作系统和所述非安全操作系统的交互，以使得所述安全操作系统和所述非安全操作系统引用彼此的服务功能以完成双系统引导。

[0011] 优选地，所述执行所述安全操作系统的至少部分引导工作包括：

[0012] 进行所述安全操作系统的待加载镜像的真实性与完整性校验；

[0013] 配置所述安全操作系统的可信资源。

[0014] 优选地，所述利用统一的服务入口实现所述安全操作系统和所述非安全操作系统的交互，以使得所述安全操作系统和所述非安全操作系统引用彼此的服务功能以完成双系统引导包括：

[0015] 加载所述非安全操作系统的至少部分服务功能；

[0016] 执行所述安全操作系统的引导工作，直到所述安全操作系统的至少部分服务功能加载完毕；其中，在执行所述安全操作系统的引导工作的过程中，所述安全操作系统通过所述统一的服务入口引用所述非安全操作系统加载的所述至少部分服务功能；

[0017] 执行所述非安全操作系统的剩余引导工作，直到所述非安全操作系统加载完毕。

[0018] 优选地，在执行所述非安全操作系统的剩余引导工作的过程中，所述非安全操作系统通过所述统一的服务入口引用所述安全操作系统加载的所述至少部分服务功能。

[0019] 优选地，所述方法还包括：

[0020] 当接收到所述非安全操作系统的调用应用指令时，根据所述调用应用指令包含的调用参数加载与所述调用参数对应的安全操作系统的对应应用。

[0021] 根据本发明实施例的第二方面，公开了一种面向终端安全的双系统引导装置，所述终端包括安全操作系统与非安全操作系统，所述安全操作系统运行于安全执行环境中，所述非安全操作系统运行于非安全执行环境中，所述装置包括：

[0022] 第一执行单元，用于执行所述安全操作系统的至少部分引导工作，在对所述安全操作系统的内存管理单元初始化之前，暂停对其引导；

[0023] 第二执行单元，用于执行所述非安全操作系统的引导工作直到所述非安全操作系统的内核加载完毕；其中，在执行所述非安全操作系统的引导工作的过程中，执行所述非安全操作系统的内存管理单元的初始化处理；

[0024] 交互引导单元，用于利用统一的服务入口实现所述安全操作系统和所述非安全操作系统的交互，以使得所述安全操作系统和所述非安全操作系统引用彼此的服务功能以完成双系统引导。

[0025] 优选地，所述第一执行单元具体用于：

[0026] 进行所述安全操作系统的待加载镜像的真实性与完整性校验；

[0027] 配置所述安全操作系统的可信资源。

[0028] 优选地，所述交互引导单元包括：

[0029] 第一加载单元，用于加载所述非安全操作系统的至少部分服务功能；

[0030] 第二加载单元，用于执行所述安全操作系统的引导工作，直到所述安全操作系统的至少部分服务功能加载完毕；其中，在执行所述安全操作系统的引导工作的过程中，所述安全操作系统通过所述统一的服务入口引用所述非安全操作系统加载的所述至少部分服务功能；

[0031] 第三加载单元，用于执行所述非安全操作系统的剩余引导工作，直到所述非安全

操作系统加载完毕。

[0032] 优选地，所述第三加载单元，具体用于在执行所述非安全操作系统的剩余引导工作的过程中，通过所述统一的服务入口引用所述安全操作系统加载的所述至少部分服务功能。

[0033] 优选地，所述装置还包括：

[0034] 应用加载单元，用于当接收到所述非安全操作系统的调用应用指令时，根据所述调用应用指令包含的调用参数加载与所述调用参数对应的安全操作系统的对应应用。

[0035] 本发明实施例的一个方面能够达到的有益效果为：在本发明实施例中，在进行面向终端安全的双系统引导时，首先执行安全操作系统的至少部分引导工作，在对所述安全操作系统的内存管理单元初始化之前，暂停对其引导，然后执行非安全操作系统的引导工作并对非安全操作系统的内存管理单元进行初始化，在加载完非安全操作系统的内核后，利用统一的服务入口实现所述安全操作系统和所述非安全操作系统的交互，以使得所述安全操作系统和所述非安全操作系统引用彼此的服务功能以完成双系统引导。本发明实施例由于先对非安全操作系统的内存管理单元初始化，再对安全操作系统的内存管理单元初始化，由于安全操作系统的执行权限高于非安全操作系统的执行权限，因此大幅减少了非安全操作系统内存管理单元操作无效或异常的现象，降低安全操作系统和非安全系统的操作冲突，提高系统的可用性并减少了整个系统的引导时间，并减少了双系统在各平台上的移植工作，避免了非安全操作系统内核的碎片化，降低了系统开销。

附图说明

[0036] 为了更清楚地说明本发明实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明中记载的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[0037] 图 1 为现有技术存在的双系统引导方法流程示意图；

[0038] 图 2 为本发明实施例提供的一种面向终端安全的双系统引导方法流程示意图；

[0039] 图 3 本发明实施例提供的又一种面向终端安全的双系统引导方法流程示意图；

[0040] 图 4 本发明实施例提供的面向终端安全的双系统引导装置示意图。

具体实施方式

[0041] 为了使本技术领域的人员更好地理解本发明中的技术方案，下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都应当属于本发明保护的范围。

[0042] 首先对本发明的思想进行阐述。

[0043] 在现有技术存在的双系统引导方法，是先引导安全操作系统等到安全操作系统完全运行后，再对非安全操作系统进行校验与引导。其具体实现可以参照图 1 所示的流程示意图，包括：

[0044] S101，在安全执行环境下 (Secure World)，安全操作系统引导初始化 (Secure Boot)。

[0045] S102，待加载镜像校验与可信资源配置 (Image Check and Trusted Configuration)。

[0046] S103，安全操作系统（又可称为可信操作系统，Trusted OS) 加载。

[0047] S104，安全操作系统应用 (Trusted OS APP) 加载。

[0048] S105，跳转到非安全执行环境 (Non-Secure World)，非安全操作系统引导初始化 (Non-Secure Boot)。

[0049] S106，富操作系统内核 (Rich OS Kernel) 加载。

[0050] S107，富操作系统 (Rich OS System) 加载。

[0051] S108，富操作系统应用 (Rich OS APP) 加载。

[0052] 发明人在实现本发明的过程中发现，现有技术存在的双系统引导方法适用于安全端采用较为简单的单进程系统场景，例如如裸机系统、ucos-II（一种可移植、可固化的、可裁剪的、占先式多任务实时内核，适用于多种微处理器、微控制器和数字处理芯片）等。然而，随着移动支付、数据隐私等市场的不断发展与成熟，在多数情况下安全端存在多个运营方，安全端本身也需要采用更高的安全隔离策略，并提供更多更方便的功能。目前业界中一个比较通用的做法是在安全端采用基于 L4 的安全微内核系统（但不局限于此），该类系统以能力集管理为核心提供一套基于硬件内存管理单元 (MMU) 的安全隔离机制，适用于上述复杂应用场景。然而，在这类双系统安全方案中，采用传统安全引导方式将会带来以下几个问题严重影响整体方案的推广与使用：

[0053] 第一，双系统在不同平台的移植工作量将大幅度增加。其主要原因是由于安全操作系统完全初始化时，需要对内存管理单元 (MMU) 进行一些 CPU 敏感指令操作，该类操作会引起与非安全端操作系统的大量 MMU 操作无效和异常。因此，在双系统移植时，需要大范围修改非安全端操作系统内核代码（如 Android kernel)，并带来了非安全端操作系统内核的版本碎片化。

[0054] 第二，双系统相互共享服务存在问题。在双系统引导的过程中，很多系统服务需要在系统引导或初始化时相互引用。例如，在安全端（即在安全执行环境下）引导及加载功能模式时可能需要非安全端（即在非安全执行环境下）提供 Socket 服务。传统引导方式只能够简单满足非安全端引用安全端服务情况，安全端在初始化过程中，由于非安全端操作系统未加载服务功能，因此不能够引用非安全端的服务功能。现有技术存在的方法不适合双系统服务相互引用情况，而随着安全端系统的不断演化，双系统服务相互引用的需求越来越多，由此带来很多问题。

[0055] 第三，现有技术存在的方法系统引导时间明显增加。其主要原因是由于在安全端系统中存在较多安全应用，全部加载比较消耗时间。

[0056] 综上所述，如何在保证安全的前提下有效解决上述问题成为了双系统安全方案中的关键。

[0057] 基于此，本发明公开了一种面向终端安全的双系统引导方法和装置，可以降低安全操作系统和非安全系统的操作冲突，降低系统开销。

[0058] 参见图 2，为本发明实施例提供的一种面向终端安全的双系统引导方法流程示意

图，所述方法应用于终端，所述终端包括安全操作系统与非安全操作系统，所述安全操作系统运行于安全执行环境中，所述非安全操作系统运行于非安全执行环境中，所述方法包括：

[0059] S201，执行所述安全操作系统的至少部分引导工作，在对所述安全操作系统的内存管理单元初始化之前，暂停对其引导。

[0060] 本发明具体实现时，可以将双系统引导工作分为两个引导阶段。在第一个引导阶段中，执行所述安全操作系统的至少部分引导工作，在对所述安全操作系统的内存管理单元初始化之前，暂停对其引导。具体实现时，第一个引导阶段从 CPU 上电开始，进行所述安全操作系统的待加载镜像的真实性与完整性校验；配置所述安全操作系统的可信资源。

[0061] 但与现有技术存在的引导方法不同的是，本发明实施例在安全操作系统（又可以称为可信操作系统）内存管理单元 MMU 初始化之前暂停对其引导，并跳转到非安全端对非安全操作系统（例如富操作系统 Rich OS）进行引导。

[0062] S202，执行所述非安全操作系统的引导工作直到所述非安全操作系统的内核加载完毕；其中，在执行所述非安全操作系统的引导工作的过程中，执行所述非安全操作系统的内存管理单元的初始化处理。

[0063] 本发明实施例中的第二阶段引导工作从对执行所述非安全操作系统的引导工作直到所述非安全操作系统的内核加载完毕开始，然后执行步骤 S203 中双系统的交互以完成双系统引导。其中，在执行所述非安全操作系统的内核加载的过程中，执行所述非安全操作系统的内存管理单元的初始化处理，这时涉及到大量的内存管理单元的操作。

[0064] 通过这种引导方法，非安全端的非安全操作系统 MMU 操作无效与异常现象大幅度减少，这是因为按照本发明所提出的引导方式，非安全端操作系统 MMU 将会先行初始化，早于安全操作系统的内存管理单元初始化过程，由于安全端可信操作系统的执行权限要高于非安全端的非安全操作系统，因此大幅减少了非安全操作系统内存管理单元 MMU 操作无效或异常的现象，降低安全操作系统和非安全系统的操作冲突，并减少了双系统在各平台上的移植工作，避免了非安全操作系统内核的碎片化，降低了系统开销。

[0065] S203，利用统一的服务入口实现所述安全操作系统和所述非安全操作系统的交互，以使得所述安全操作系统和所述非安全操作系统引用彼此的服务功能以完成双系统引导。

[0066] 具体实现时，步骤 S203 可以包括：

[0067] S203A，加载所述非安全操作系统的至少部分服务功能。

[0068] 其中，加载的所述非安全操作系统的至少部分服务功能为在执行所述安全操作系统的引导工作的过程中被所述安全操作系统引用的服务功能。具体地，在非安全执行环境下，非安全操作系统将初始化一些安全操作系统需要使用的必要服务功能，例如 socket 等，以供安全操作系统引导过程中使用。然后，系统将跳转到安全执行环境，继续加载安全操作系统。

[0069] S203B，执行所述安全操作系统的引导工作，直到所述安全操作系统的至少部分服务功能加载完毕；其中，在执行所述安全操作系统的引导工作的过程中，所述安全操作系统通过所述统一的服务入口引用所述非安全操作系统加载的所述至少部分服务功能。

[0070] 在执行所述安全操作系统的引导工作的过程中，所述安全操作系统通过所述统一

的服务入口引用所述非安全操作系统的加载的部分服务功能。安全操作系统和非安全操作系统将可能存在若干次交互直到安全操作系统全部加载完毕。其中，所述安全操作系统的至少部分服务功能为所述非安全操作系统引导过程中需要使用的功能。

[0071] S203C，执行所述非安全操作系统的剩余引导工作，直到所述非安全操作系统加载完毕。

[0072] 在加载完安全操作系统后，将执行所述非安全操作系统的剩余引导工作，直到所述非安全操作系统加载完毕。类似地，在非安全操作系统引导与加载过程中，可能需要使用到安全操作系统的服务功能，非安全操作系统和安全操作系统将可能存在若干次交互直到非安全操作系统全部加载完毕。需要说明的是，在执行所述安全操作系统的引导工作的过程中，加载所述安全操作系统的至少部分服务功能；在执行所述非安全操作系统的剩余引导工作的过程中，所述非安全操作系统通过所述统一的服务入口引用所述安全操作系统的所述至少部分服务功能。

[0073] 由此，即完成了双系统的引导工作。在本发明具体实现时，在第二阶段引导中，采用双系统相互共享服务功能的方法可以更有效地划分安全操作系统与非安全操作系统的功能边界，即将高安全、低开销的功能操作放到安全执行环境（如 SE、加解密），将低安全，高开销的功能操作放到非安全执行环境（如 Socket），两端通过统一的服务入口进行相互调用，从而在保证安全的前提下，缩减安全端开销与系统规模，减少安全认证成本。

[0074] 进一步地，本发明实施例提供的方法还可以包括：当接收到所述非安全操作系统的调用应用指令时，根据所述调用应用指令包含的调用参数加载与所述调用参数对应的安全操作系统的对应应用。

[0075] 需要说明的是，在本发明实施例提供的双系统引导方法中，与现有技术不同的是，在双系统引导过程中，并不先加载安全操作系统的应用程序，而是采用按需加载的方式，延迟加载安全应用，所述案应用的加载工作不包含在双系统引导过程中。当非安全操作系统通过安全操作系统的应用入口发送调用应用指令以进行应用调用时，安全操作系统将实时解析非安全操作系统的调用应用指令包含的调用参数，按需加载与所述调用参数对应的被调用安全应用。安全应用执行完完毕后随即挂起，同时返回非安全操作系统并等待下一次调用。通过采用本发明所提出的安全应用按需延迟加载方式，可以有效减少双系统引导时间，增加用户体验，并且由于每次调用采用按需加载的方式，有效的减少了安全操作系统的内存开销。

[0076] 参见图 3，本发明实施例提供的又一种面向终端安全的双系统引导方法流程示意图。

[0077] 图 3 所示的实施例将以非安全操作系统为富操作系统为例进行说明。

[0078] S301，在安全执行环境下 (Secure World)，安全操作系统引导初始化 (Secure Boot)。

[0079] S302，进行所述安全操作系统的待加载镜像的真实性与完整性校验以及配置所述安全操作系统的可信资源 (Image Check and Trusted Configuration)。

[0080] S303，跳转到非安全执行环境 (Non-Secure World)，非安全操作系统引导初始化 (Non-Secure Boot)。

[0081] S304，富操作系统内核 (Rich OS Kernel) 加载。

[0082] S305，富操作系统服务功能加载，其中所述服务功能为安全操作系统引导与加载所需功能。

[0083] S306，跳转到安全执行环境，执行安全操作系统的引导工作，在此过程中，引用富操作系统的服务功能。

[0084] S307，加载安全操作系统的服务功能，所述服务功能可以被非安全操作系统引用，用于所述非安全操作系统的引导与加载工作。

[0085] S308，富操作系统 (Rich OS System) 加载。

[0086] S309，富操作系统应用 (Rich OS APP) 加载，向安全操作系统发送调用应用指令。

[0087] S310，按照所述调用指令按需加载安全应用。

[0088] 在本发明的一种应用场景中，采用 ARM v7A 处理器平台，所述富操作系统 (Rich OS) 为 Android 4.2 系统，所述安全操作系统 (Trusted OS) 采用 L4Microkernel 系统，能够在保证安全的前提下达到以下效果：

[0089] 1. 由于采用本发明提出的两阶段引导方式，非安全操作系统的 MMU 操作冲突明显减少，富操作系统内核修改工作大大减少，有效减少了双系统整体移植工作量。

[0090] 2. 基于本发明提出的两阶段引导方式以及新型的双系统相互共享服务方法，明确了安全端与非安全端功能的划分边界，有效地缩减了安全端开销与系统规模，减少安全认证成本。

[0091] 3. 由于本发明中安全应用采用延迟按需加载方式，有效地减少了双系统引导时间，增加了用户体验，并有效的减少了安全端内存开销，缩减安全成本。

[0092] 参见图 4，本发明实施例提供的面向终端安全的双系统引导装置示意图。

[0093] 一种面向终端安全的双系统引导装置 400，所述终端包括安全操作系统与非安全操作系统，所述安全操作系统运行于安全执行环境中，所述非安全操作系统运行于非安全执行环境中，所述装置 400 包括：

[0094] 第一执行单元 401，用于执行所述安全操作系统的至少部分引导工作，在对所述安全操作系统的内存管理单元初始化之前，暂停对其引导。

[0095] 第二执行单元 402，用于执行所述非安全操作系统的引导工作直到所述非安全操作系统的内核加载完毕；其中，在执行所述非安全操作系统的引导工作的过程中，执行所述非安全操作系统的内存管理单元的初始化处理。

[0096] 交互引导单元 403，用于利用统一的服务入口实现所述安全操作系统和所述非安全操作系统的交互，以使得所述安全操作系统和所述非安全操作系统引用彼此的服务功能以完成双系统引导。

[0097] 优选地，所述第一执行单元具体用于：

[0098] 进行所述安全操作系统的待加载镜像的真实性与完整性校验；

[0099] 配置所述安全操作系统的可信资源。

[0100] 优选地，所述交互引导单元包括：

[0101] 第一加载单元，用于加载所述非安全操作系统的至少部分服务功能；

[0102] 第二加载单元，用于执行所述安全操作系统的引导工作，直到所述安全操作系统的至少部分服务功能加载完毕；其中，在执行所述安全操作系统的引导工作的过程中，所述安全操作系统通过所述统一的服务入口引用所述非安全操作系统加载的所述至少部分服

务功能；

[0103] 第三加载单元，用于执行所述非安全操作系统的剩余引导工作，直到所述非安全操作系统加载完毕。

[0104] 优选地，所述第三加载单元，具体用于在执行所述非安全操作系统的剩余引导工作的过程中，通过所述统一的服务入口引用所述安全操作系统加载的所述至少部分服务功能。

[0105] 优选地，所述装置还包括：

[0106] 应用加载单元，用于当接收到所述非安全操作系统的调用应用指令时，根据所述调用应用指令包含的调用参数加载与所述调用参数对应的安全操作系统的对应应用。

[0107] 需要说明的是，上述各单元的功能可对应于图 2 至图 3 所详细描述的上述方法的处理步骤，于此不再赘述。需要说明的是，由于对方法实施例进行详细的阐述，对装置实施例的描述较为简单，本领域技术人员可以理解的是，可以参照方法实施例构造本发明的装置实施例。本领域技术人员在不付出创造性劳动下获取的其他实现方式均属于本发明的保护范围。

[0108] 本领域技术人员可以理解的是，以上对方法和装置实施例进行了示例性说明，以上不视为对本发明的限制，本领域技术人员在不付出创造性劳动下获得的其他实现方式均属于本发明的保护范围。

[0109] 需要说明的是，在本文中，诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来，而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。本发明可以在由计算机执行的计算机可执行指令的一般上下文中描述，例如程序模块。一般地，程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本发明，在这些分布式计算环境中，由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中，程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

[0110] 本说明书中的各个实施例均采用递进的方式描述，各个实施例之间相同相似的部分互相参见即可，每个实施例重点说明的都是与其他实施例的不同之处。尤其，对于装置实施例而言，由于其基本相似于方法实施例，所以描述得比较简单，相关之处参见方法实施例的部分说明即可。以上所描述的装置实施例仅仅是示意性的，其中所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。本领域普通技术人员在不付出创造性劳动的情况下，即可以理解并实施。以上所述仅是本发明的具体实施方式，应当指出，对于本技术领域的普通技术人员来说，在不脱离本发明原理的前提下，还可以做出若干改进和润饰，这些改进和润饰也应视为本发明的保护范围。

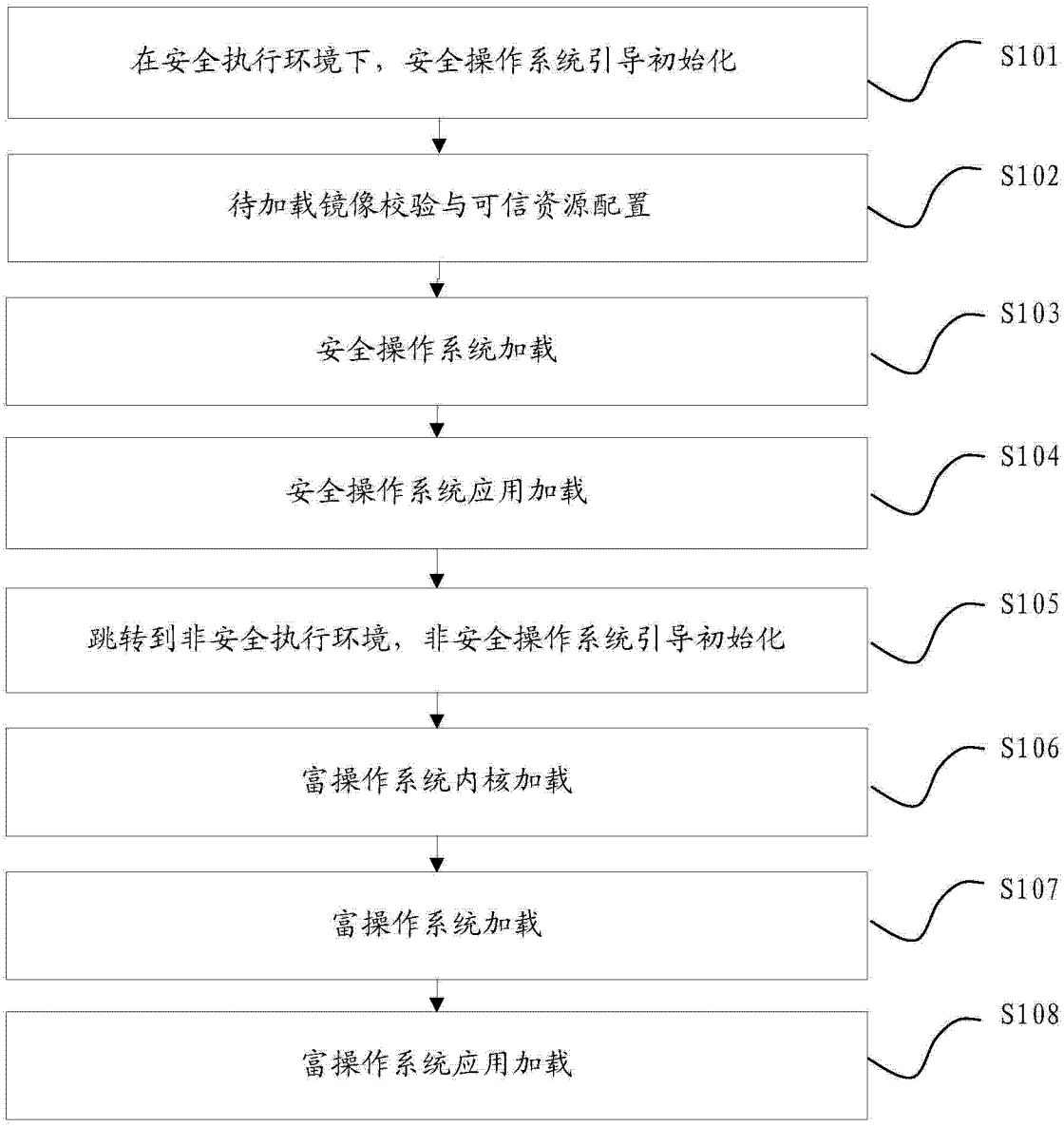

在安全执行环境下，安全操作系统引导初始化
S101
待加载镜像校验与可信资源配置
S102
安全操作系统加载
S103
安全操作系统应用加载
S104
跳转到非安全执行环境，非安全操作系统引导初始化
S105
富操作系统内核加载
S106
富操作系统加载
S107
富操作系统应用加载
S108

图 1

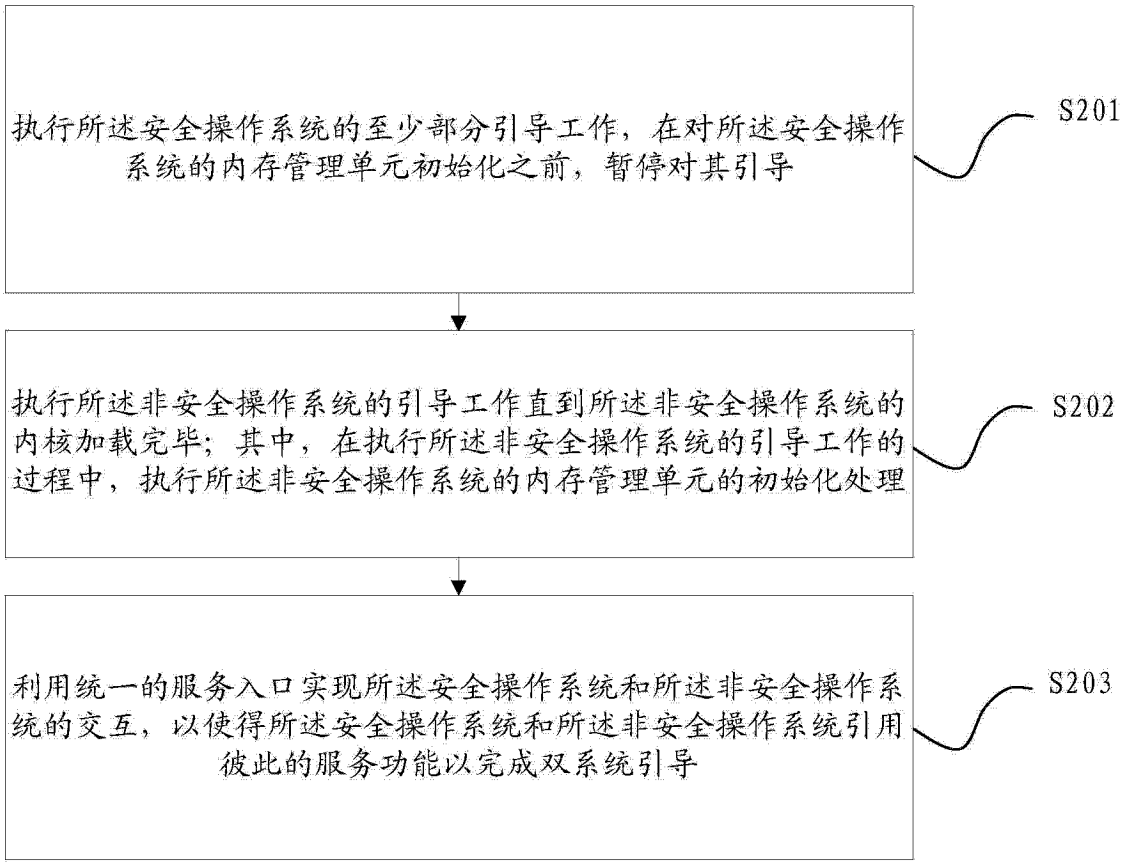

执行所述安全操作系统的至少部分引导工作，在对所述安全操作系统的内存管理单元初始化之前，暂停对其引导
S201
执行所述非安全操作系统的引导工作直到所述非安全操作系统的内核加载完毕；其中，在执行所述非安全操作系统的引导工作的过程中，执行所述非安全操作系统的内存管理单元的初始化处理
S202
利用统一的服务入口实现所述安全操作系统和所述非安全操作系统的交互，以使得所述安全操作系统和所述非安全操作系统引用彼此的服务功能以完成双系统引导
S203

图 2

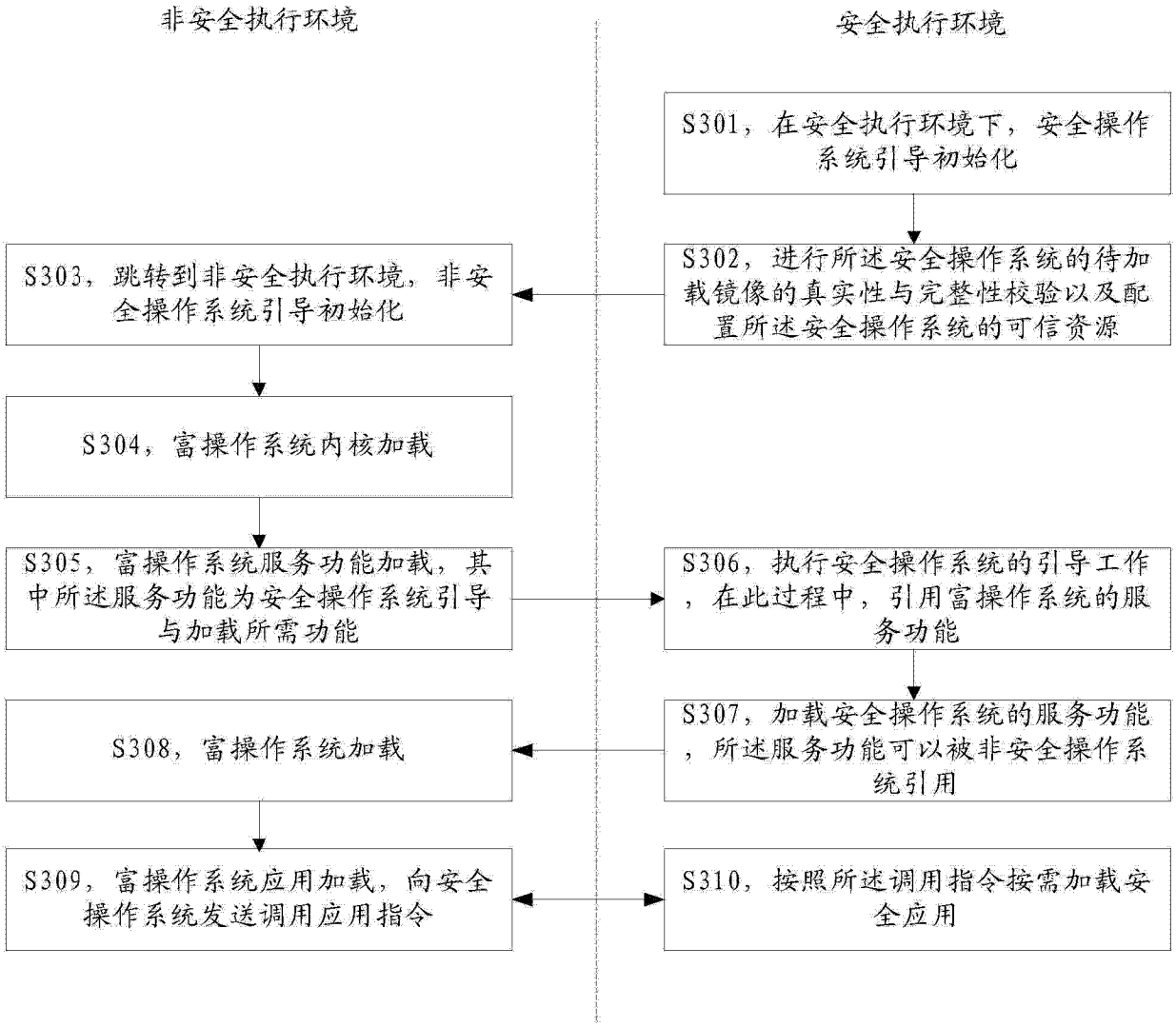

非安全执行环境
安全执行环境
S301，在安全执行环境下，安全操作系统引导初始化
S302，进行所述安全操作系统的待加载镜像的真实性与完整性校验以及配置所述安全操作系统的可信资源
S303，跳转到非安全执行环境，非安全操作系统引导初始化
S304，富操作系统内核加载
S305，富操作系统服务功能加载，其中所述服务功能为安全操作系统引导与加载所需功能
S306，执行安全操作系统的引导工作，在此过程中，引用富操作系统的服务功能
S307，加载安全操作系统的服务功能，所述服务功能可以被非安全操作系统引用
S308，富操作系统加载
S309，富操作系统应用加载，向安全操作系统发送调用应用指令
S310，按照所述调用指令按需加载安全应用

图 3

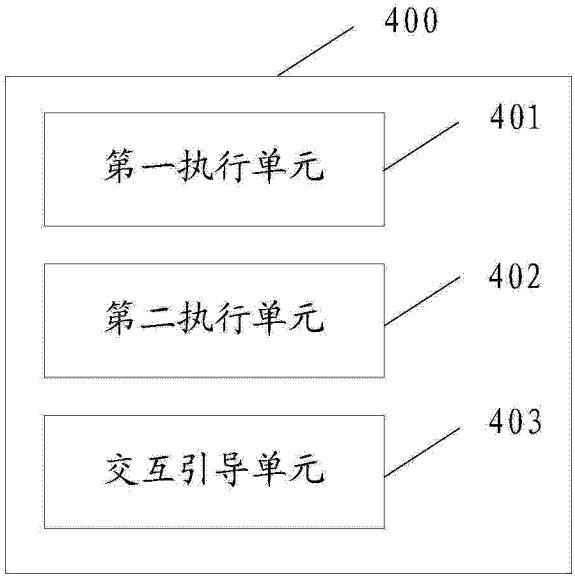


图 4