(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 116204884 A
(43)申请公布日 2023.06.02

(21)申请号 202111461175.9

(22)申请日 2021.11.30

(71)申请人 华为技术有限公司
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72)发明人 黄振强 瓦伦丁·马尼亚 朱健伟

(74)专利代理机构 北京中博世达专利商标代理有限公司 11274
专利代理师 申健

(51)Int.Cl.
G06F 21/57(2013.01)
G06F 12/1009(2016.01)

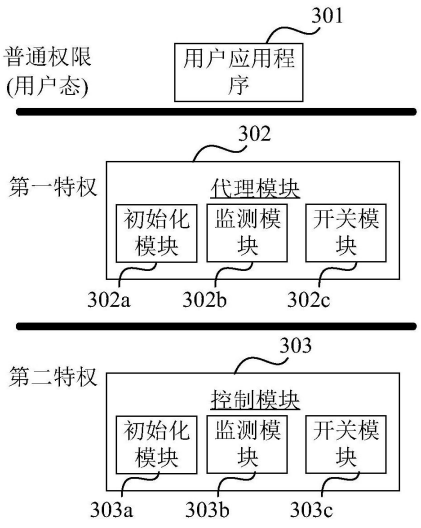
权利要求书2页 说明书21页 附图17页

(54)发明名称

内核保护方法、装置及系统

(57)摘要

一种内核保护方法、装置及系统，涉及安全技术领域，可以降低内核遭受攻击的概率，提升系统的安全性。该方法应用于电子设备，所述方法包括：工作在第一特权，并检测到页表修改命令；所述第一特权包括第一特权；所述页表修改命令用于修改目标页表中的访问权限数据；所述目标页表是内核相关的页表；所述第一特权是内核特权；由所述第一特权切换至第二特权，并在所述第二特权下，根据所述页表修改命令判断是否对所述目标页表进行修改；所述第二特权的权限高于所述第一特权；若确定对所述目标页表进行修改，则修改所述目标页表中的访问权限数据。



CN 116204884 A

1.一种内核保护方法,其特征在于,应用于电子设备,所述方法包括:

工作在第一特权,并检测到页表修改命令;所述第一特权包括第一特权;所述页表修改命令用于修改目标页表中的访问权限数据;所述目标页表是内核相关的页表;所述第一特权是内核特权;

由所述第一特权切换至第二特权,并在所述第二特权下,根据所述页表修改命令判断是否对所述目标页表进行修改;所述第二特权的权限高于所述第一特权;

若确定对所述目标页表进行修改,则修改所述目标页表中的访问权限数据。

2.根据权利要求1所述的方法,其特征在于,所述内核相关的页表包括如下任一种或多种页表:用于映射内核代码段的页表,用于映射驱动代码段的页表,用于映射数据段的页表;

根据所述页表修改命令判断是否对所述目标页表进行修改,包括:

如果所述目标页表为用于映射内核代码段的页表,且所述页表修改命令为预设命令,则确定修改所述目标页表。

3.根据权利要求1或2所述的方法,其特征在于,所述方法还包括:在所述第一特权下,创建第一物理内存、第一页表以及第二页表;

其中,所述第一页表和所述第二页表存储在所述第一物理内存中;第一页表为用于映射所述第一物理内存的页表,第二页表是内核相关的页表。

4.根据权利要求1或2所述的方法,其特征在于,所述第一特权下,所述第一物理内存的访问权限为只读。

5.根据权利要求3所述的方法,其特征在于,所述第一特权下,所述第一页表不包括写入权限,所述第二特权下,所述第一页表包括写入权限。

6.根据权利要求2所述的方法,其特征在于,所述预设命令包括insn命令。

7.一种内核保护装置,其特征在于,包括:

处理单元,工作在第一特权,并检测到页表修改命令;所述第一特权包括第一特权;所述页表修改命令用于修改目标页表中的访问权限数据;所述目标页表是内核相关的页表;所述第一特权是内核特权;由所述第一特权切换至第二特权,并在所述第二特权下,根据所述页表修改命令判断是否对所述目标页表进行修改;所述第二特权的权限高于所述第一特权;若确定对所述目标页表进行修改,则修改所述目标页表中的访问权限数据。

8.根据权利要求7所述的装置,其特征在于,所述内核相关的页表包括如下任一种或多种页表:用于映射内核代码段的页表,用于映射驱动代码段的页表,用于映射数据段的页表;

根据所述页表修改命令判断是否对所述目标页表进行修改,包括:

如果所述目标页表为用于映射内核代码段的页表,且所述页表修改命令为预设命令,则确定修改所述目标页表。

9.根据权利要求7或8所述的装置,其特征在于,所述处理单元,还用于在所述第一特权下,创建第一物理内存、第一页表以及第二页表;

其中,所述第一页表和所述第二页表存储在所述第一物理内存中;第一页表为用于映射所述第一物理内存的页表,第二页表是内核相关的页表。

10.根据权利要求7或8所述的装置,其特征在于,所述第一特权下,所述第一物理内存

的访问权限为只读。

11.根据权利要求9所述的装置,其特征在于,所述第一特权下,所述第一页表不包括写入权限,所述第二特权下,所述第一页表包括写入权限。

12.根据权利要求8所述的装置,其特征在于,所述预设命令包括insn命令。

13.一种电子设备,其特征在于,包括:

一个或多个处理器;

存储器;

以及一个或多个计算机程序,其中所述一个或多个计算机程序存储在所述存储器上,当所述计算机程序被所述一个或多个处理器执行时,使得所述电子设备执行如权利要求1-6中任一项所述的方法。

14.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质包括计算机程序或指令,当所述计算机程序或指令在计算机上运行的情况下,使得所述计算机执行如权利要求1-6中任一项所述的方法。

15.一种计算机程序产品,其特征在于,所述计算机程序产品包括:计算机程序或指令,当所述计算机程序或指令在计算机上运行的情况下,使得所述计算机执行如权利要求1-6中任一项所述的方法。

内核保护方法、装置及系统

技术领域

[0001] 本申请涉及安全技术领域，尤其涉及内核保护方法、装置及系统。

背景技术

[0002] 服务器上大量的应用程序以普通权限运行在用户空间。当进行网络通信或者硬件资源访问时，则需要提权到具有更高权限(内核特权)的内核空间进行。所以，服务器内核的安全性对整个服务器的健康运行至关重要

[0003] 目前，一旦应用获得第一特权，则系统信任该应用，攻击者可以任意篡改，达到攻击窃取信息的目的，风险大。因此，亟待提出一种保护内核免受入侵的方案。

发明内容

[0004] 本申请提供内核保护方法、装置及系统，可以降低内核遭受攻击的概率，提升系统的安全性。

[0005] 为了实现上述目的，本申请实施例提供了以下技术方案：

[0006] 第一方面提供内核保护方法，应用于电子设备，所述方法包括：

[0007] 工作在第一特权，并检测到页表修改命令；所述第一特权包括第一特权；所述页表修改命令用于修改目标页表中的访问权限数据；所述目标页表是内核相关的页表；所述第一特权是内核特权；

[0008] 由所述第一特权切换至第二特权，并在所述第二特权下，根据所述页表修改命令判断是否对所述目标页表进行修改；所述第二特权的权限高于所述第一特权；

[0009] 若确定对所述目标页表进行修改，则修改所述目标页表中的访问权限数据。

[0010] 可选的，第一特权为内核特权。

[0011] 本申请实施例的技术方案，使得内核特权下无法直接修改内核中关键内存的访问权限，如果需要修改关键内存的权限，则需切换至更高特权(比如第二特权)完成修改，因此，能够降低在内核特权下遭受攻击的概率。

[0012] 关键内存权限：页表具有很多权限，比较重要的如读、写、执行权限。而不同页表对应的权限，在抵御内核入侵上所起的作用也不同。本专利中的关键内存权限是指抵御内核漏洞利用的页表权限，包括内核代码段和驱动代码段的写入权限，rop-pool页表的写入权限，以及数据段的执行权限。

[0013] 在一种可能的设计中，所述内核相关的页表包括如下任一种或多种页表：用于映射内核代码段的页表，用于映射驱动代码段的页表，用于映射数据段的页表；

[0014] 根据所述页表修改命令判断是否对所述目标页表进行修改，包括：

[0015] 如果所述目标页表为用于映射内核代码段的页表，且所述页表修改命令为预设命令，则确定修改所述目标页表。

[0016] 在一种可能的设计中，所述方法还包括：在所述第一特权下，创建第一物理内存、第一页表以及第二页表；

[0017] 其中,所述第一页表和所述第二页表存储在所述第一物理内存中;第一页表为用于映射所述第一物理内存的页表,第二页表是内核相关的页表。

[0018] 在一种可能的设计中,所述第一特权下,所述第一物理内存的访问权限为只读。

[0019] 在一种可能的设计中,所述第一特权下,所述第一页表不包括写入权限,所述第二特权下,所述第一页表包括写入权限。

[0020] 在一种可能的设计中,所述预设命令包括insn命令。

[0021] 如下,对本申请实施例涉及的部分技术术语进行简单介绍:

[0022] 重要页表:能影响内核运行逻辑的页表,如映射内核代码段的页表,映射各驱动模块代码段的页表,本专利中映射rop-pool物理内存的页表,以及部分数据段页表。在具有页表权限优先级的芯片架构上,部分数据段页表指映射数据段的高层级页表。而其他芯片架构上,部分数据段页表指映射需要被监测的数据段的页表。专利实现者可根据实际情况,如数据段已采用其他技术进行保护,则可不需要监测,或者部分监测。

[0023] 专用页表池:在内核初始化初期开辟的一段物理内存,用于保存内核页表。在内核初始化过程中,将内核代码段和数据段等的页面映射放入专用页表池,并将专用页表池自身的页面映射也放入其中;内核初始化后期,将专用页表池的写入权限去除。为了更直观反应其意义,本专利中也用rop-pool对专用页表池进行标记:rop-pool表示在内核运行时,该页表池在内核态(或者说是特权A模式)呈现为只读,如果要修改,则需要陷入更高权限完成。页表专用池是本专利方案中的一个核心控制数据块。

[0024] 自身映射(self-mapping)机制:内核初始化完成后,去除重要页表的写入权限,后续内核特权下无法恢复该写入权限,从而保障重要页表数据无法被修改。

[0025] 高特权合法性判决机制:位于高特权的控制程序对内核的代理程序下发的页表权限的修改操作进行合法性判断,合法则修改,不合法则拒绝。

[0026] 在本申请实施例的技术方案中,内核初始化过程中,代理程序触发控制会去除内核重要页表的写入权限,后续在内核特权(比如特权A或B)下无法被恢复,保障内核中关键内存权限无法被篡改。内核运行过程中,代理程序会将内核中关键内存权限的修改交于控制程序执行,控制程序进行合法性判断后再选择是否执行,从而阻止非法修改并保障系统正常运行。

[0027] 此外,非内存数据(如寄存器等)在内核中无法禁止被修改,需要在控制程序中做周期性检测(告警)。除了寄存器,专利实现者可根据实际需要,亦可在控制程序中对DMA页表数据,rop-pool页表数据,内核代码段等重要内存数据做周期性检测比如检测异常变更。

[0028] 需要说明的是,本申请实施例的双特权技术要求CPU具有不同权限等级,目前通用的arm和x86芯片皆支持该功能。

[0029] 进一步的,开源内核在创建页表时,随机申请物理内存进行页表数据存放,不便于对其进行管理和控制。为了能统一管理重要页表,开辟统一的第一物理内存,第一物理内存用于存储重要页表。

[0030] 目前,存在如下攻击方式,是直接修改硬盘文件系统上的内容,比如,假设某个KO还没加载到内存时,则通过替换为文件系统中的KO进行篡改;还有一种攻击方式,可以直接替换内核。本申请实施例中,为了降低系统的安全风险,可以通过安全启动确保文件系统的安全性,使得文件系统中改变的任何东西,在加载时无法通过签名,加载失败,如此用户就

可以知道系统出现问题。可见，通过安全启动方式，可以确保文件系统、内核段等重要内存的安全性。

[0031] 第二方面提供一种内核保护装置，包括：

[0032] 处理单元，工作在第一特权，并检测到页表修改命令；所述第一特权包括第一特权；所述页表修改命令用于修改目标页表中的访问权限数据；所述目标页表是内核相关的页表；所述第一特权是内核特权；由所述第一特权切换至第二特权，并在所述第二特权下，根据所述页表修改命令判断是否对所述目标页表进行修改；所述第二特权的权限高于所述第一特权；若确定对所述目标页表进行修改，则修改所述目标页表中的访问权限数据。

[0033] 在一种可能的设计中，所述内核相关的页表包括如下任一种或多种页表：用于映射内核代码段的页表，用于映射驱动代码段的页表，用于映射数据段的页表；

[0034] 根据所述页表修改命令判断是否对所述目标页表进行修改，包括：

[0035] 如果所述目标页表为用于映射内核代码段的页表，且所述页表修改命令为预设命令，则确定修改所述目标页表。

[0036] 在一种可能的设计中，所述处理单元，还用于在所述第一特权下，创建第一物理内存、第一页表以及第二页表；

[0037] 其中，所述第一页表和所述第二页表存储在所述第一物理内存中；第一页表为用于映射所述第一物理内存的页表，第二页表是内核相关的页表。

[0038] 在一种可能的设计中，所述第一特权下，所述第一物理内存的访问权限为只读。

[0039] 在一种可能的设计中，所述第一特权下，所述第一页表不包括写入权限，所述第二特权下，所述第一页表包括写入权限。

[0040] 在一种可能的设计中，所述预设命令包括insn命令。

[0041] 第三方面，本申请提供一种电子设备，该电子设备具有实现如上述任意方面及其中任一种可能的实现方式中的内核保护方法的功能。该功能可以通过硬件实现，也可以通过硬件执行相应的软件实现。该硬件或软件包括一个或多个与上述功能相对应的模块。

[0042] 第四方面，本申请提供一种计算机可读存储介质，包括计算机指令，当计算机指令在电子设备上运行时，使得电子设备执行如上述任意方面及其中任一种可能的实现方式中任一项的内核保护方法。

[0043] 第五方面，本申请提供一种计算机程序产品，当计算机程序产品在电子设备上运行时，使得电子设备执行如任意方面及其中任一种可能的实现方式中任一项的内核保护方法。

[0044] 第六方面，提供一种电路系统，电路系统包括处理电路，处理电路被配置为执行如上述任意方面及其中任一种可能的实现方式中的内核保护方法。

[0045] 第七方面，本申请实施例提供一种芯片系统，包括至少一个处理器和至少一个接口电路，至少一个接口电路用于执行收发功能，并将指令发送给至少一个处理器，当至少一个处理器执行指令时，至少一个处理器执行如上述任意方面及其中任一种可能的实现方式中的内核保护方法。

附图说明

[0046] 图1为本申请实施例提供的MMU机制的运行原理；

[0047] 图2A、图2B为本申请实施例提供的页表的查询机制；

[0048] 图3为本申请实施例提供的系统架构的示意图；

[0049] 图4为本申请实施例提供的内核保护装置的结构示意图；

[0050] 图5为本申请实施例提供的内核保护装置的软件结构示意图；

[0051] 图6为本申请实施例提供的内核保护方法的流程示意图；

[0052] 图7A、图7B为本申请实施例提供的页表的示意图；

[0053] 图8A-图8F为本申请实施例提供的页表的示意图；

[0054] 图9-图12为本申请实施例提供的内核保护方法的流程示意图；

[0055] 图13-图15为本申请实施例提供的内核保护装置的软件架构示意图；

[0056] 图16为本申请实施例提供的装置的示意图；

[0057] 图17为本申请实施例提供的芯片系统的示意图。

具体实施方式

[0058] 首先，对本申请实施例涉及的技术术语进行介绍：

[0059] 内存访问机制：内核通过内存管理单元(memory management unit，MMU)，将虚拟内存地址(virtual address，VA)映射为物理内存地址(physical address，PA)。具体的，如图1所示，程序运行时，程序对应的地址为虚拟地址，中央处理单元(central processing unit，CPU)通过MMU将虚拟地址解析为物理地址，然后通过MMU对相应物理地址的内存进行寻址访问。

[0060] 本申请实施例中，内存可以是各种形式内存，比如可以但不限于是双速率(double data rate，DDR)内存。

[0061] 页表(page)以及页表在内存寻址中的作用(页表的作用之一)：MMU机制通过多级页表中的最低级别页表存放虚拟地址和物理地址之间的映射关系数据。

[0062] 其中，多级页表包含顶级页表(也可称为一级页表)和多级子页表。可选的，顶级页表包括但不限于页目录(page directory，PGD)。子页表包括但不限于页表条目(page table，PTE)。顶级页表的地址存储在页表寄存器中，子页表的地址存储在其上一级子页表中。最低级别的子页表中存储虚拟内存地址以及对应的物理内存地址。示例性的，如图2A所示，顶级页表中存储二级子页表的地址，二级子页表中存储三级子页表(为图2A例子中的最低级子页表)的地址，以此类推。三级子页表中存储物理内存地址B对应的基址。后续，可以根据虚拟地址A中的偏移量以及物理地址B对应的基址，计算物理地址B的地址。

[0063] 图2A示出了一种内存寻址的方案，CPU在通过MMU解析虚拟地址A时，MMU先访问页表寄存器，获取顶级页表的物理地址，并根据顶级页表的地址寻址访问相应内存地址，获取顶级页表。之后MMU逐级解析子级页表，在最低级别子页表中获取待访问虚拟地址A对应的物理地址B。

[0064] 页表权限技术(页表在内存保护中的作用(页表的作用之二))：在一些方案中，为了实现对内核运行过程中内核相关内存的完整性保护，可以采用页表权限技术，对内核相关的内存空间(内存区域)的访问权限进行控制。页表权限技术，通过在内存对应的页表中加入该内存的访问权限(如RWX)，来对漏洞利用进行防御，防止内核代码被篡改和数据段注入执行等攻击。其中，R表示读取权限，即可以对相应的内存进行读取操作。W表示写入权限，

即可以对相应的内存进行写入操作。X表示执行权限，即相应内存上的指令代码可以被执行；如果没有执行权限，则对应的代码指令不能被执行。示例性的，如图2A所示，三级子页表中的访问权限数据是R、W权限，表示允许对地址B对应的内存进行读、写操作。再如图2B所示，三级子页表中的访问权限数据是R、W、X权限，表示允许对地址B对应的内存进行读、写、执行操作。

[0065] 其中，内核相关的内存，可以是用于存储内核相关代码、内核相关数据的内存。比如可以包括但不限于数据段所在内存、代码段所在内存。代码段包括但不限于内核代码段、驱动代码段。内核代码段包括但不限于内核vmlinuz。

[0066] 在本申请实施例中，驱动还可以称为驱动模块、驱动文件等，名称并不构成对驱动的功能的限制。驱动代码段，指的是驱动的代码段。可选的，驱动模块包括但不限于ko模块。可选的，驱动代码段包括但不限于ko代码段。

[0067] 在本申请实施例中，用于映射内核相关内存的页表，可以称为内核相关页表。内核相关页表包括但不限于：用于映射内核代码段的页表、用于映射驱动代码段的页表、用于映射数据段的页表。

[0068] 在一些方案中，针对不同的内存空间，可以有不同的访问权限。示例性的，如图2B所示，为了防止非法入侵者在数据段中注入非法代码，并执行非法代码，导致系统逻辑篡改，可以在三级子页表中将物理地址B对应的数据段的执行权限去除(即该部分数据仅具有读写权限)，使得即便在数据段所在内存注入了非法代码，也无法执行该非法代码，进而保证系统安全。

[0069] 页表映射，可以包括页表对物理地址和虚拟地址之间的映射，也可以包括页表对物理地址与相应访问权限之间的映射。

[0070] 可以看出，页表中的访问权限数据是较为重要的数据，其能够通过控制一段内存的读、写、执行权限，来尽可能提升该段内存的安全。

[0071] 本申请实施例中，将用于存储数据段对应的虚拟地址和物理地址，以及该数据段对应的访问权限的页表，称为用于映射数据段的页表，进一步的，还可简称为数据段的页表。示例性的，如图2A所示的三级子页表即为数据段的页表。

[0072] 类似的，将用于映射内核代码段的页表，简称为内核代码段的页表。将用于映射驱动代码段的页表，简称为驱动代码段的页表。

[0073] 页表权限优先级：是指高层级页表中的权限会覆盖低层级页表中的权限。比如，顶级页表中的权限设置为不可执行，则该顶级页表包含的各子级页表即使包含执行权限，也无法执行。示例性的，如果如图2A所示的顶级页表中地址B的权限设置为不可执行，那么，即便三级子页表中地址B对应的权限设置为可执行，后续，地址B对应内存空间中的数据也不允许执行。

[0074] 在一些方案中，CPU的不同运行特权层级下，有不同的页表映射机制。可以通过不同特权层级的页表映射机制控制不同特权层级下对某些内存的访问权限。可选的，在普通权限下，内存A具有只读权限，即只允许对内存A进行读取，不允许写入，在第一特权下，内存A具有读取、写以及执行权限。示例性的，如图2A示出了普通权限下的页表映射。如图2B示出了第一特权下的页表映射。

[0075] 漏洞：可以理解为可以被攻击者利用的系统或者程序代码的缺陷，可以是代码的

bug,也可以是系统设计的缺陷等。

[0076] 第一特权:某些情况下,攻击者可以利用漏洞(包括但不限于数组越界等漏洞),进行系统提权,即由普通权限提升到第一特权(包括但不限于root权限)。在攻击者获得第一特权后,被攻击系统基于信任等原因,不再对攻击者的诸如页表修改操作进行安全校验,使得攻击者能够通过重新建立页表或页表重映射(double-mapping)等方式篡改页表,包括但不限于修改页表中的权限数据,比如修改页表中某段内存(比如内核关联的内存)对应的读、写、执行权限,进而实现对系统内核的攻击。示例性的,攻击者在获得第一特权之后,可以绕过系统的防御机制,在图2B所示三级子页表A中的权限数据中增加执行(X)权限。如此一来,后续,攻击者可以在地址B对应的内存空间中注入恶意代码并执行该恶意代码,实现对系统内核的攻击。

[0077] 内核入侵:在一些方式中,攻击者可以通过篡改内核代码的实现逻辑,入侵内核,对系统造成直接的破坏。在另一些方式中,攻击者通过向内核注入恶意代码,潜伏入系统,在不被管理员或者用户发现的情况下,不断窃取重要数据,这类恶意代码往往不会对系统造成直接的破坏,但是对用户具有更大的危害性,可以造成用户信息泄露或者财产损失等重大危害。

[0078] 内核代码段的篡改:在一些攻击方式中,在获得第一特权后,攻击者可以获得对内核代码段进行写入操作的权限,这样一来,攻击者可以通过篡改内核代码段中的内核代码,篡改系统的实现逻辑,实现攻击目的。

[0079] 数据段注入执行:在一些攻击方式中,在获得第一特权后,攻击者可以获得对数据段进行写入、执行操作的权限,攻击者可以在数据段中注入恶意代码,并执行该恶意代码,以实现攻击目的。

[0080] 为了防止上述内核入侵或系统潜伏问题,本申请实施例提供一种内核保护方法,考虑到页表中存储有内核相关内存的一些访问权限,因此,可以通过保护页表来保护这些内核相关内存,进而实现对系统内核的保护。

[0081] 在本申请的技术方案中,数据段、代码段不同时具有写入权限和执行权限。以数据段为例,若允许对数据段进行写入操作,则不允许对数据段进行执行操作,或者说,禁止对数据段添加允许执行的权限。如此一来,即便攻击者进行提权获得第一特权,其在数据段注入了恶意代码,其也会因为被禁止为数据段添加执行权限,而无法运行注入的恶意代码。如此,可以尽可能的阻止恶意代码被执行

[0082] 以代码段为例,若允许对代码段进行执行操作,则不允许对代码段进行写入操作,或者,禁止对代码段添加允许写入的权限。如此一来,由于不允许在第一特权下对代码段进行写入操作,因此,可尽可能避免内核代码遭受篡改。

[0083] 本申请实施例提供的内核保护方法可应用于需要进行内核保护的系统中。如图3示出了一种示例性的系统架构,该系统包括一个或多个电子设备。电子设备可以云侧设备或端侧设备。云侧设备包括但不限于是服务器(比如图3所示服务器300)。端侧设备包括但不限于是终端(比如笔记本电脑100、台式电脑200)。当然,图3仅是举例,本申请实施例对电子设备的具体形态不进行限制。其中,本申请实施例涉及的电子设备可以是手机、电脑等运行有内核,并需要进行内核保护的设备。

[0084] 参见图4,为本申请实施例提供的电子设备的一种可能的结构示意图。例如,该设

备可以是终端,或者终端中的部件,或服务器,或服务器中的部件。在一种可能的实现方式中,电子设备可以包括存储器201、处理器202、通信器件203和总线204。存储器201、处理器202以及通信器件203通过总线204相互连接。

[0085] 其中,存储器201可用于存储数据、软件程序以及模块,主要包括存储程序区和存储数据区,存储程序区可存储操作系统、至少一个功能所需的应用程序等,存储数据区可存储该装置的使用时所创建的数据等。在本申请实施例中,上述操作系统可以包括通用操作系统(比如,Linux®系统)或安全操作系统。安全操作系统比如可以是可信执行环境操作系统(trusted execution environment operating system,TEE OS)。上述至少一个功能所需的应用程序可以包括基于各种操作系统的内核的程序、驱动模块的程序。上述存储数据区存储的数据可以包括但不限于内核的数据。

[0086] 在本申请的一些实施例中,处理器(比如CPU)中可包括MMU,或者,MMU可以为独立于CPU之外的功能模块。

[0087] 在一些可行的实施例中,存储器201可以是软盘,硬盘如内置硬盘和移动硬盘,磁盘,光盘,磁光盘如只读光盘(compact disc read-only memory,CD_ROM)、数字视盘(digital video disc read-only memory,DVD_ROM),非易失性存储设备如随机存取存储器(random access memory,RAM)、只读存储器(read-only memory,ROM)、可编程只读存储器(programmable read-only memory,PROM)、可擦写可编程只读存储器(erasable programmable read-only memory,EPROM)、电可擦写可编程只读存储器(electrically erasable programmable read-only memory,EEPROM)、闪存、或者技术领域内所公知的任意其他形式的存储介质。

[0088] 处理器202用于对设备的动作进行控制管理,比如,通过运行或执行存储在存储器201内的软件程序和/或模块,以及调用存储在存储器201内的数据,执行设备的各种功能和处理数据。其中,处理器202可以包括安全元件(Secure element,SE)、中央处理单元、数字信号处理器、人工智能(artificial intelligence,AI)处理器、微处理器、或微控制器等的至少一个。

[0089] 可选的,处理器还选择性包括专用集成电路、现场可编程门阵列或者其他可编程逻辑器件、晶体管逻辑器件、硬件加速器或者其任意组合。可选的,处理器也可以是实现计算功能的组合。进一步的,在本申请实施例中,处理器202可选择性包括其他硬件加速器,该组合可用于支持设备执行本申请中内核保护的功能,其中,内核保护方法的具体过程可以参见下述方法实施例中的描述。

[0090] 通信器件203用于支持电子设备与其他设备进行通信。通信器件203可以是接收器、接收电路、发送器、发送电路、具有收发功能的收发器、收发电路等。比如,通信器件203是基于近场通信(near field communica-tion,NFC)的收发电路,或蓝牙等其他模式的近距离通信器件。

[0091] 总线204可以是外设部件互连标准(peripheral component interconnect,PCI)总线,或者扩展工业标准结构(extended industry standard architecture,EISA)总线等。总线可以分为地址总线、数据总线、控制总线等。为便于表示,图4中仅用一条粗线表示,但并不表示仅有一根总线或一种类型的总线。

[0092] 在本申请实施例的一些实现方式中,处理器202本身可以视为一个用于实现内核

保护功能的装置(可简称为内核保护装置),其可作为运算核心实现内核保护功能。或者,处理器202可位于一个内核保护装置,其与内核保护装置中的其他部件可共同实现内核保护功能。该内核保护装置可以是一个或多个芯片、芯片组,或可以选择性运行软件来工作。

[0093] 图4仅是本申请实施例的电子设备的一种示例性结构。在另一些实施例中,电子设备还可以有更多或更少的组件,或具有其他组件布局方式,或将图4中的某些组件进行拆分,或将图4所示某些组件集成。本申请实施例对电子设备的实现形态不做限制。

[0094] 图5为本申请实施例提供的一种处理器202上运行的软件的架构示意图。其中,处理器202,可用于运行用户应用程序301、代理模块302以及控制模块303。其中,处理器202具有不同的运行权限,各类应用程序在各自对应的运行权限下运行。

[0095] 作为一种可能的实现方式,用户应用程序301通常运行在用户权限(或称普通权限)。用户应用程序301可以通过调用内核提供的访问硬件和资源的接口,来访问硬件和相应资源。通常,在用户权限下,用户应用程序301能够访问的硬件和资源有限,即只能访问用户权限所允许访问的硬件和资源,以降低用户应用程序301对系统的篡改和攻击风险。

[0096] 内核运行在第一特权(第一特权)下,第一特权比用户权限高。因此,在第一特权下运行的内核能够比用户权限下运行的用户应用程序301访问更多的硬件和资源。需要说明的是,某些情况下,用户应用程序301也可以陷入第一特权运行。比如,在执行系统调用时,用户应用程序301可以陷入第一特权中运行。

[0097] 代理模块302,运行在第一特权下,第一特权的权限高于用户权限。作为一种可能的实现方式,代理模块302,包括初始化子模块302a、监测子模块302b、开关子模块302c。

[0098] 其中,在内核初始化过程中,代理模块中的初始化子模块302a,可以与控制模块303中的初始化子模块303a配合,完成内核初始化中的一些配置。可选的,在内核初始化过程中,初始化子模块302a可以将内核的相关信息发送给初始化子模块303a。可选的,内核的相关信息包括但不限于如下任一项或多项信息:重要页表的信息、初始化子模块303a可以根据内核的相关信息,

[0099] 在内核运行过程中,监测子模块302b,可以与控制模块中的检测模块303b配合,实现监测功能。监测功能,包括但不限于如下任一项或多项:监测页表修改命令是否合法,根据页表修改命令修改相应页表。其中,页表修改命令用于修改目标页表中的访问权限数据。目标页表是内核相关的页表。

[0100] 比如,在检测到页表修改命令时,将相应的页表修改命令转发给控制模块303中的监测子模块303b,由监测子模块303b根据页表修改命令判断是否进行页表修改操作,若页表修改命令合法,则修改相应页表,若页表修改命令不合法,则拒绝修改相应页表。

[0101] 在内核运行过程中,开关子模块302c,可以在检测到内核加载(插入)驱动时,向控制模块303中的开关子模块303c发送监测功能关闭命令。监测功能关闭命令用于指示/请求/通知控制模块303停止对驱动代码段相关的页表修改命令的合法性监测。如此,当接收到对驱动代码段相关的页表修改命令(比如修改页表中ko模块的访问权限的命令)之后,控制模块303(比如控制模块303中的监测子模块303b)不再对该页表修改命令的合法性进行判断,而是直接根据该页表修改命令对相应页表进行修改。

[0102] 开关子模块302c,用于支持动态加载内核模块或者加载热补丁操作等。

[0103] 开关子模块302c,还可以在驱动插入完成时,向控制模块303中的开关子模块303c

下发监测功能恢复命令。监测功能恢复命令用于指示/请求/通知控制模块303恢复对驱动代码段的监测功能。

[0104] 控制模块303，运行在第二特权下，第二特权的权限高于第一特权。作为一种可能的实现方式，控制模块303，包括初始化子模块303a、监测子模块303b、开关子模块303c。

[0105] 其中，初始化子模块303a，可以从初始化子模块302a接收内核的相关信息，并根据内核的相关信息xx。

[0106] 监测子模块303b，可以从监测子模块302b接收页表修改命令，并判断该页表修改命令是否合法，若合法，则根据页表修改命令对相应页表修改，若非法，则拒绝修改相应页表。

[0107] 监测子模块303b，还可以在内核初始化完成后，检测内核相关的寄存器。当检测到篡改寄存器的操作后，向代理模块上报告警。监测子模块303b检测内核相关的寄存器，可以是周期性检测，或者按照策略检测。比如，当检测到某些特定事件时，触发检测内核相关寄存器。本申请实施例对监测子模块303b的检测时机以及检测条件不做限制。

[0108] 开关子模块303c，可以从开关子模块302c接收监测功能关闭命令，并根据监测功能关闭命令，停止对驱动代码段相关的页表修改命令的合法性监测。还可以从开关子模块302c接收监测功能恢复命令，并根据监测功能恢复命令，恢复对驱动代码段相关的页表修改命令的合法性监测。

[0109] 可选的，先启动控制模块303，再启动代理模块302，即控制模块303先于代理模块302工作，如此，能够尽可能提升控制模块303的工作可靠性，使得控制模块303能够接收来自代理模块302的数据，并根据这些数据对页表的修改进行控制、管理。

[0110] 可选的，代理模块302的合法性、ko模块、内核段等的合法性，由安全启动保障。如此，能够确保代理模块302、ko模块在硬盘上不被篡改或者替换。

[0111] 内核代码段区域的完整性通过安全启动保障，内核加载完成后，该专利方案已经开始起保护作用。所以，认为内核代码段区域的命令为合法命令。

[0112] 在一些实施例中，代理模块还可称为代理程序或其他名称，但这不构成对代理模块的功能的限制。类似的，控制模块还可以称为控制程序或其他名称，但这并不构成对控制模块的功能的限制。

[0113] 图5仅是本申请实施例的电子设备的一种示例性软件架构。在另一些实施例中，电子设备还可以有更多或更少的模块，或具有其他模块布局方式，或将图5中的某些模块进行拆分，或将图5所示某些模块集成。本申请实施例对电子设备上运行的软件的具体形态、布局不做限制。

[0114] 本申请的说明书以及附图中的术语“第一”和“第二”等是用于区别不同的对象，或者用于区别对同一对象的不同处理。“第一”、“第二”等字样可以对功能和作用基本相同的相同项或相似项进行区分。例如，第一设备和第二设备仅仅是为了区分不同的设备，并不对其先后顺序进行限定。本领域技术人员可以理解“第一”、“第二”等字样并不对数量和执行次序进行限定，并且“第一”、“第二”等字样也并不限定一定不同。

[0115] “至少一个”是指一个或者多个，

[0116] “多个”是指两个或两个以上。

[0117] “和/或”，描述关联对象的关联关系，表示可以存在三种关系，例如，A和/或B，可以

表示:单独存在A,同时存在A和B,单独存在B的情况,其中A,B可以是单数或者复数。字符“/”一般表示前后关联对象是一种“或”的关系。“以下至少一项(个)”或其类似表达,是指的这些项中的任意组合,包括单项(个)或复数项(个)的任意组合。例如,a,b,或c中的至少一项(个),可以表示:a,b,c,a-b,a-c,b-c,或a-b-c,其中a,b,c可以是单个,也可以是多个。

[0118] 此外,本申请的描述中所提到的术语“包括”和“具有”以及它们的任何变形,意图在于覆盖不排他的包含。例如包含了一系列步骤或单元的过程、方法、系统、产品或设备没有限定于已列出的步骤或单元,而是可选地还包括其他没有列出的步骤或单元,或可选地还包括对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0119] 需要说明的是,本申请实施例中,“示例性的”或者“例如”等词用于表示作例子、例证或说明。本申请实施例中被描述为“示例性的”或者“例如”的任何实施例或设计方案不应被解释为比其它实施例或设计方案更优选或更具优势。确切而言,使用“示例性的”或者“例如”等词旨在以具体方式呈现相关概念。

[0120] 本申请实施例中,在提及某个消息或指令用于xx时,指的是该消息或指令等可以用于实现某个功能(比如指示修改页表),但不不局限于专门用于实现该功能。

[0121] 本申请描述的系统架构及业务场景是为了更加清楚的说明本申请的技术方案,并不构成对于本申请提供的技术方案的限定,本领域普通技术人员可知,随着系统架构的演变和新业务场景的出现,本申请提供的技术方案对于类似的技术问题,同样适用。

[0122] 如下结合方法实施例来说明上述各软件、硬件模块的工作原理。

[0123] 本申请实施例提供一种内核保护方法,适用于内核初始化流程。如图6示出了一种内核初始化的流程,该内核初始化流程包括如下步骤:

[0124] S101、代理模块确认控制模块是否正常运行。

[0125] 本申请实施例中,控制模块运行在高于第一特权(即第一特权)的第二特权下。控制模块用于判断相应页表修改命令是否合法,并在页表修改命令合法时,进行页表修改操作。因此,本申请实施例的技术方案能够正常实施的前提是确保控制模块正常工作。

[0126] 作为一种可能的实现方式,S101可实现为:S101a、在内核初始化过程中,代理模块向控制模块发送第一消息;S101b、控制模块向代理模块发送第二消息。其中,第一消息用于确认控制模块是否正常运行。第二消息用于指示控制模块正常运行。

[0127] 可以理解,若代理模块接收到来自控制模块的第二消息,则说明控制模块正常运行,那么,继续执行本申请实施例的后续流程,比如继续执行下述步骤S102,S103,在第一特权下进行页表映射。

[0128] 反之,若代理模块没有接收到来自控制模块的第二消息,或第二消息错误(即S101b的执行出现问题),则意味着控制模块没有正常运行,那么,代理模块将退出工作,本申请实施例的后续步骤不再执行,内核按照现有技术的流程执行相应操作。

[0129] S102、代理模块开辟第一物理内存。

[0130] 其中,第一物理内存用于存储用于映射重要内存的页表。用于映射重要内存的页表可称为重要页表。重要内存是保证内核相关的内存,这些重要内存对内核安全影响程度较大。内核运行的相关内存包括但不限于数据段所在内存,代码段所在内存。其中,代码段包括但不限于内核代码段和驱动代码段。在下述实施例中,将使用专用页表池的概念代替第一物理内存。

[0131] 相应的，重要页表包括但不限于部分数据段的页表、内核代码段的页表、驱动代码段的页表，这些页表可称为第一页表。

[0132] 在本申请实施例中，重要页表还可以包括用于映射专用页表池(rop-pool)的页表，该类页表可称为第二页表。第二页表存储有专有页表池的虚拟地址以及对应的物理地址，还存储有专有页表池对应的读、写、执行权限。

[0133] 作为一种可能的实现方式，控制模块正常运行的情况下，在内核启动(即内核初始化)初期，代理模块开辟一段物理内存(即第一物理内存)作为专用页表池，专用页表池可以用来存储多个上述重要页表。

[0134] S103、代理模块创建重要页表，并将重要页表存入专用页表池。

[0135] 示例性的，如图7A示例性的示出了代理模块创建的专用页表池。如图7A所示，在为专用页表池创建相关页表的过程中，代理模块将一些重要页表(比如部分数据段的页表、内核代码段的页表、驱动代码段的页表、专用页表池的页表)均存入该专用页表池中。其中，在专用页表池的页表中存储有整个专用页表池的虚拟地址以及对应的物理地址，以及存储有整个专用页表池的读、写、执行权限。

[0136] 本申请实施例中，将上述专用页表池存储第二页表(即映射该专用页表池这段物理内存的页表)的机制，称为自身映射(self-mapping)机制。如此，能够实现页表映射该页表所在物理内存(即专用页表池)的功能，进而通过该页表控制专用页表池，以控制专用页表池中的全部重要页表。仍以图7A所示，将用于映射专用页表池的页表A存储在该专用页表池中，如此，可以通过页表A所在的物理内存控制整块专用页表池的物理内存，即可以通过页表A统一控制专用页表池这段内存上的重要页表(包括但不限于代码段和数据段)的读、写、执行权限。

[0137] 需要说明的是，在内核初始化的初期，由于存在向内核相关内存中加载数据、代码等操作，就需要内核相关内存有相应的写入、执行权限。示例性的，图7A示出了内核初始化的初期，重要页表中第一页表的权限数据。其中，驱动代码段的页表B中的权限为读、写、执行，以保证在内核初始化初期，能够向驱动代码段所在内存中插入并执行驱动代码。内核代码段的页表C中的权限为读、写、执行，能够保证在内核初始化初期，将内核代码加载到内核代码段所在内存，并执行该载入的内核代码。数据段的页表D中的权限为读、写，能够保证在内核初始化初期，向数据段所在内存写入相应数据。

[0138] 图7A还示出了内核初始化的初期，重要页表中的第二页表，即页表A中的访问权限。可以看出，页表A的访问权限为只读，意味着，后续在第一特权下，无法对第一物理内存上的内容进行修改，进而无法在第一特权下对重要页表B-D中的访问权限数据进行修改。如此，能够提升系统内核的安全性。

[0139] 上述S102、S103，是内核初始化初期，第一特权对应的页表映射过程。在该页表映射过程中，在各个重要页表中写入相应的访问权限数据、地址映射数据。

[0140] S104、第二特权下的页表映射过程。

[0141] 在一些实施例中，在内核初始化初期，第二特权下也有相应页表映射机制。可选的，由控制模块实现第二特权下的页表映射。作为一种可能的实现方式，控制模块在映射代码段和数据段的页表时，可以复用第一特权下的页表映射，无需重建页表。在映射专用页表池的页表时，可以重新创建专用页表池的页表。可选的，第二特权下不存在自身映射，专用

页表池的页表不再存储在专用页表池中。示例性的，图8A示出了第二特权下的页表映射机制，第二特权下，控制模块为专用页表池创建新的页表A1，该专用页表池的页表A1不再存储在专用页表池中。且，数据段的页表D、内核代码段的页表C、驱动代码段的页表B是复用第一特权下的页表。

[0142] 值得注意的是，第二特权下，专用页表池的页表A1中的访问权限数据需要包括写入(R)权限，以便于后续控制模块能够在页表修改命令合法时，对专用页表池中的页表进行修改。

[0143] 本申请实施例中，为了提升内核相关内存的安全性，可以在内核相关内存中相关数据、代码加载完成之后，内核相关内存不再需要执行或写入的访问权限时，对第一特权下重要页表中的写入和/或执行权限进行去除，使得重要内存不会同时具有写入以及执行的访问权限，以防止内核运行过程中，攻击者在获得第一特权后，对重要页表映射的重要内存进行篡改。比如，在内核初始化后期，去除代码段在第一特权下的写入权限，以防止攻击者在获得第一特权后篡改内核相关代码。内核相关代码包括但不限于内核代码段、驱动代码段、数据段。

[0144] 上文已提到，由于专用页表池对应的页表A中的访问权限为只读，因此，第一特权下无法对专用页表池进行写入操作，也就无法修改专用页表池中的各重要页表，即无法在第一特权下对重要页表中的写入或执行权限进行去除。作为一种可能的实现方式，为了对第一特权下重要页表中的写入或执行权限进行去除，以防止内核运行过程中，攻击者通过诸如篡改重要页表的访问权限数据达到攻击目的，在内核初始化后期，由运行在第二特权的控制模块对部分重要页表中的访问权限数据进行修改，也就是，在第二权限下对重要页表中的写入或执行权限进行去除。其中，第二特权高于第一特权。后续，在内核运行阶段，即便攻击者获得第一特权，也无法在第一特权下篡改重要页表中的权限数据。下述步骤S105-S1010为控制模块修改重要页表(包括修改重要页表中的访问权限数据)的具体实现。

[0145] S105、代理模块向控制模块发送内核相关的内存信息。

[0146] 可选的，内核相关的内存信息包括但不限于如下任一项或多项信息：内核代码段所在内存的地址、驱动代码段所在内存的地址、数据段所在内存的地址、专用页表池所在内存的地址信息、内核相关的重要寄存器的地址。

[0147] 可选的，专用页表池所在内存的地址信息包括：专用页表池中各第一页表所在内存的地址。

[0148] S106、控制模块存储内核相关的内存信息。

[0149] 后续，在内核运行过程中，控制模块可以根据内核相关的内存信息，进行合法性判断和周期性检测。

[0150] 其中，合法性判断，用于判断第一特权下对页表的修改是否合法。周期性检测，用于检测非页表控制的存储空间(比如寄存器)的安全性。合法性判断、周期性检测的具体实现方式可参见下述实施例。

[0151] S107、代理模块向控制模块发送第三消息。

[0152] 其中，第三消息用于通知或请求或指示控制模块去除代码段(包括内核代码段和/或驱动代码段)在第一特权下的写入权限。意味着，后续，在第一特权下，不允许对代码段进行写入操作，从而能够提升第一特权下的系统安全性。

[0153] S108、控制模块根据第三消息，去除代码段在第一特权下的写入权限。

[0154] 本申请实施例中，示例性的，如图8A所示，第二特权下，控制模块可以为专用页表池这块内存重新创建页表A1，并将该指向专用页表池的页表A1存储在专用页表池之外。也即，第二特权下，页表映射机制可以不存在自身映射。

[0155] 作为一种可能的实现方式，如图8A所示，页表A1(第二特权下的页表)中的访问权限包括读、写。意味着，后续，可以对专用页表池这块内存进行写入操作，以便后续在检测到合法的页表修改命令的情况下，在第二特权下对专用页表池中的页表进行修改，以实现或修复系统的功能。

[0156] 示例性的，以去除驱动代码段的写入权限为例，如图8A所示，在从代理模块接收第三消息之后，查询页表B1所在内存(专用页表池)对应的页表A1，专用页表池在第二特权下对应的页表A1具有写入权限，意味着，运行在第二特权下的控制模块能够对专用页表池进行写入操作。对专用页表池进行写入操作包括但不限于对专用页表池中的各页表(包括但不限于页表B、C、D)进行修改。比如，控制模块可以将图8A所示驱动代码段对应的页表B中的写入(W)权限去除，得到图8B所示驱动代码段对应的页表B1，页表B1中的权限不包括写入权限，表示不允许对驱动代码段进行写入操作，以防止攻击者在第一特权下对内核的驱动代码进行篡改。可以理解，由于第二特权下驱动代码段对应的页表、第一特权下驱动代码段对应的页表存储在同一块内存，或者说，第二特权下驱动代码段对应的页表与第一特权下驱动代码段对应的页表是同一页表，因此，当第二特权下驱动代码段对应的页表被修改后，第一特权下驱动代码段对应的页表也相应发生变化。图8B示出了内核初始化后期，经第二特权下的控制模块修改后，内核权限下驱动代码段的页表B1的一种示例。类似的，图8B示出了内核初始化后期，经第二特权下的控制模块修改后，内核权限下内核代码段的页表C1的一种示例。

[0157] 在去除代码段在第一特权下的写入权限之后，由于代码段不同时具有写入权限和执行权限，因此，能够尽可能降低篡改并执行篡改后的内核代码所导致的系统安全风险问题。

[0158] S109、代理模块向控制模块发送第四消息。

[0159] 相应的，控制模块从代理模块接收第四消息。其中，第四消息用于指示或通知或请求控制模块去除数据段在第一特权下的执行权限。

[0160] S1010、控制模块根据第四消息，去除数据段在第一特权下的执行权限。

[0161] 内核运行过程中，阻止其后续被恢复。通过步骤一的操作，在内核态没有改写专用页表池的权限，即无法恢复专用页表池的写权限，从而保证了其中的内核页表数据无法被修改。

[0162] 本申请实施例还提供一种内核保护方法，适用于内核运行过程中。如下分场景对内核运行过程中的内核保护方法进行介绍。

[0163] 场景一:修改页表场景

[0164] 在内核运行过程中，内核可能检测到修改页表的操作，修改页表的操作可能来自攻击者或合法用户。比如，攻击者为了篡改内核代码而修改页表，或者，合法用户为了加载一些功能模块(比如KO驱动模块)而修改页表。为了避免攻击者通过修改页表攻击系统，本申请实施例提供一种内核保护方法，在检测到第一特权下的页表修改操作时，需要对页表

修改操作是否合法进行判断，只有在确定页表修改操作合法时，才执行对相应页表的修改。具体的，如图9所示，该内核保护的方法包括如下步骤：

[0165] S201、代理模块检测到页表修改命令。

[0166] 其中，代理模块运行在第一特权下。页表修改命令用于指示修改目标页表(或称待修改页表)。修改目标页表，包括但不限于修改目标页表中目标内存对应的访问权限数据。示例性的，如图8B所示，代理模块检测到在页表B1中为驱动代码段(存储在地址B中)添加写入权限的页表修改命令。

[0167] S202、代理模块判断目标内存对应的目标页表(待修改页表)是否属于专用页表池，若是，则执行下述步骤S203，若否，则执行下述步骤S204。

[0168] 作为一种可能的实现方式，代理模块判断目标页表的存储地址，若目标页表的存储地址在专用页表池的地址范围内，则确定目标页表属于专用页表池，否则，确定目标页表不再专用页表池内。

[0169] S203、代理模块向控制模块发送页表修改命令。

[0170] 可选的，页表修改命令携带待修改访问权限的物理内存的地址信息。

[0171] 上文已指出，运行在第一特权下的代理模块无法直接修改重要页表，修改重要页表的操作交由运行在第二特权下的控制模块完成。具体的，作为一种可能的实现方式，内核运行过程中，发生页表修改时，代理模块会判断待修改的页表是否在专用页表池内。在一些情况下，若待修改页表属于专用页表池，则意味着待修改页表是重要页表(对内核安全的影响程度较大)，那么，为了保障内核的安全性，代理模块向运行在第二特权的控制模块下发页表修改命令，由具有更高特权的控制模块判断是否修改相应重要页表。

[0172] 示例性的，如图8B所示，代理模块检测到对页表B1的页表修改命令，并确定页表B1存储在专用页表池内(属于重要页表)。那么，为了提升内核的安全性，代理模块向控制模块转发该页表修改命令，由控制模块判断该页表修改命令是否合法。

[0173] S204、代理模块修改目标页表。

[0174] 在另一些情况下，内核运行过程中，代理模块检测到页表修改命令之后，若确定待修改页表不属于专用页表池，则意味着待修改页表不是重要页表，即便修改该待修改页表，通常也不会对系统造成损害，那么，代理模块可以按照现有技术的实现逻辑，直接对待修改页表进行修改。

[0175] S205、控制模块判断页表修改命令是否合法。若合法，则执行下述步骤S206，若非法，则执行下述步骤S207。

[0176] S206、控制模块根据页表修改命令修改目标页表。

[0177] 在一些情况下，控制模块从代理模块接收页表修改命令之后，若确定页表修改命令是合法命令，则意味着该修改重要页表的命令通常来自合法用户的操作，通常不会对系统造成攻击风险，那么，控制模块根据页表修命令对目标页表进行修改，以实现系统的相应功能。

[0178] 示例性的，在需要为内核加载热补丁，以修复系统功能的场景中，控制模块从代理模块接收页表修改命令之后，确定该命令是向内核代码中加载热补丁的合法命令(比如insn命令)，则控制模块根据该命令修改内核代码段对应的页表，使得内核代码段具有写入的访问权限，以便于正常实现向内核代码段中加载热补丁的操作。

[0179] S207、控制模块拒绝修改目标页表。

[0180] 在一些情况下，控制模块从代理模块接收页表修改命令之后，若确定该页表修改命令不是合法命令，则意味着该修改重要页表的命令通常来自非法用户的操作，那么，控制模块拒绝修改目标页表。

[0181] S208、控制模块向代理模块发送拒绝修改响应。

[0182] 可选的，若确定拒绝修改相应重要页表，则控制模块可以向代理模块发送拒绝修改响应。

[0183] 如下对控制模块判断页表修改命令是否合法的具体实现做介绍。

[0184] 作为一种可能的实现方式，控制模块从代理模块接收页表修改命令之后，若待修改页表所映射的地址在内核代码段对应的地址范围内，则说明待修改页表是用于映射内核代码段的页表。此种情况下，为了防止非法用户通过篡改页表进而篡改内核代码，只有该页表修改命令是预设命令时，控制模块才确定该页表修改命令是合法命令。

[0185] 可选的，预设命令包括但不限于insn。insn命令用于加载热补丁。

[0186] 示例性的，控制模块从代理模块接收insn命令，该命令指示需要修改图8B所示的页表C1，为地址C这块物理内存(内核代码段所在内存)添加写入权限。由于该insn命令是预设命令，则控制模块确定该insn命令合法，并在页表C1中为地址C对应的内存(内核代码段所在内存)添加写入权限，得到如图8C所示的新页表C2。

[0187] 再示例性的，控制模块从代理模块接收页表修改命令，该命令指示需要修改图8B所示的页表C1，为地址C这块物理内存(内核代码段所在内存)添加写入权限。如果该页表修改命令是insn之外的其他命令，则控制模块将该页表修改命令视为非法命令，拒绝修改页表C1，以免因修改页表C1导致系统遭受攻击。

[0188] 作为一种可能的实现方式，根据从代理模块接收的页表修改命令，若待修改页表所映射的内存地址是专用页表池本身，则控制程序确定页表修改命令是非法命令，相应的，不允许修改页表。示例性的，控制模块从代理模块接收页表修改命令，该命令指示需要修改图8A所示地址A对应物理内存的访问权限。由于地址A对应的物理内存为专用页表池所在物理内存。那么，控制模块将该页表修改命令视为非法命令，不予执行。

[0189] 作为一种可能的实现方式，若根据页表修改命令，待修改页表所映射的地址在数据段地址范围内，则说明待修改页表是用于映射数据段的页表。那么，为了避免数据段注入执行以及其他类似攻击，控制模块不允许数据段同时具有写入权限和执行权限。即，一旦检测到能够致使数据段同时具有写入权限和执行权限的页表修改命令，控制模块将该页表修改命令视为非法命令。示例性的，在一些场景中，控制模块从代理模块接收页表修改命令，该命令指示控制模块在如图8D所示的页表D1中增加执行权限，即指示为地址D对应的物理内存(数据段所在内存)添加执行权限。控制模块拒绝在页表D1中的访问权限数据中增加执行权限，以避免数据段同时具有写入和执行权限，进而能够降低系统遭受数据段注入执行攻击的风险。

[0190] 作为一种可能的实现方式，若根据页表修改命令，待修改页表所映射的物理地址属于驱动代码段地址范围，则控制模块判断当前是否开启监测功能。若控制模块当前开启了监测功能，则控制模块执行合法性判断，即需要判断页表修改命令是否能够致使驱动代码段同时具有写入和执行权限。若页表修改命令能够致使驱动代码段同时具有写入和执行

权限，则控制模块将该页表修改命令视为非法命令，且不响应该页表修改命令。若页表修改命令不会致使驱动代码段同时具有写入和执行权限，则控制模块将该页表修改命令视为合法命令。

[0191] 示例性的，如图8E所示，控制模块从代理模块接收页表修改命令，且该命令指示为地址B这块内存(即驱动代码段所在内存)添加写入(W)权限，即需要在页表B2中添加写入权限。如果此时控制模块开启了监测功能，则控制模块会对该页表修改命令进行合法性判断，经判断，在页表B2中添加写入权限这一页表修改操作会导致驱动代码段同时具有写入和执行权限，增加注入执行攻击的风险，因此，控制模块拒绝在页表B2中的访问权限数据中增加写入权限。

[0192] 反之，若控制模块当前没有开启监测功能，则控制模块不执行合法性判断，即不对页表修改命令是否合法进行判断，而是直接将来自代理模块的页表修改命令视为合法命令，并执行页表修改命令所指示的操作，修改目标页表。

[0193] 示例性的，如图8F，控制模块从代理模块接收页表修改命令A，且该命令指示为地址B这块内存(即驱动代码段所在内存)添加写入(W)权限，即需要在页表B2中添加写入权限。如果此时控制模块未开启监测功能(监测功能关闭)，则控制模块不对该页表修改命令A进行合法性判断，而是直接根据该页表修改命令A，在页表B2中添加写入权限。

[0194] 作为一种可能的实现方式，内核代码段的合法性可以由安全启动过程保障，控制模块的开关子模块可以判断页表修改命令所在地址是否属于内核代码段，进而判断是否需要关闭监测功能。如果页表修改命令位于内核代码段区域，则意味着页表修改命令是安全的，那么，控制模块可以暂时关闭监测功能。这样一来，可以在关闭监测功能之后，根据页表修改命令对相应页表进行修改，从而具有访问相应内存的相应访问权限，支持正常的系统功能(比如支持在驱动代码段中加载ko模块)。

[0195] 示例性，如图8F所示，控制模块从代理模块接收页表修改命令A，该页表修改命令A存储在内核代码段中。控制模块根据该页表修改命令A的存储地址，确定开该页表修改命令A不是由恶意程序下发的，那么，控制模块可以关闭监测功能，以便能够修改页表B2，以支持系统的正常功能(比如通过修改页表B2加载ko模块)。

[0196] 反之，如果页表修改命令不属于内核代码段，则无法确保页表修改命令的合法性，此种情况下，不关闭监测功能，即继续维持监测功能。示例性的，如图8E所示，控制模块从代理模块接收页表修改命令A，该页表修改命令A存储在内核代码段之外的内存X中。由于内核代码段之外的该内存X没有经安全启动的校验，因此不能直接确保该内存X中的代码是否安全，那么，控制模块保持开启监测功能，继续对页表修改命令A的合法性进行检测。示例性的，如图8E所示，对于欲在页表B2中添加写入权限的页表修改命令A，控制模块确定该页表修改命令A能够致使驱动代码段同时具有写入和执行权限，则不响应该页表修改命令A，即拒绝修改页表B2，以降低系统遭受注入执行攻击的风险。

[0197] S209、代理模块输出告警信息。

[0198] 可选的，代理模块从控制模块接收拒绝修改响应之后，可以输出告警信息，以提示用户当前可能存在非法的内核入侵操作，用户可以据此告警信息对系统进行保护。或者，若在一段时间内，代理模块没有从控制模块接收确认消息(用于确认对相应重要页表进行修改)，则代理模块可以输出告警信息。

[0199] 作为一种可能的实现方式，代理模块通过显示驱动调用显示屏显示告警信息。

[0200] S210、控制模块检测针对重要寄存器的相关操作。

[0201] 针对重要寄存器的相关操作，包括但不限于：对重要寄存器的变更操作、

[0202] S211、在检测到对重要寄存器的异常变更操作时，向代理模块发送告警信息。

[0203] 可选的，内核运行过程中，对于页表机制的监测盲区，比如一些重要寄存器，可以进行周期性或响应于特定事件度量，在检测到异常变更重要寄存器时，控制模块向代理模块发送告警信息。代理模块可以通过显示驱动显示该告警信息，以提示用户当前系统可能存在风险。

[0204] 可选的，度量周期的具体数值，可根据实际情况进行调整，或者，度量周期可以根据CPU的两种特权之间切换的频率确定。本申请实施例度量周期的具体设置方式以及具体数值，不做限制。

[0205] 在另一些实施例中，控制模块还可以对专用页表池、内核代码段、驱动代码段、数据段、等进行周期性或响应于事件的度量，并对相应的操作。

[0206] 场景二：新建页表场景

[0207] 内核运行过程中，系统有时会根据需求创建新的页表。在本申请实施例中，当检测到开辟大内存需要创建新页表时，会判断待创建的目标页表是否属于重要页表。如果待创建的目标页表属于重要页表，则由控制模块创建该目标页表，并将目标页表加入专用页表池，以便后续对重要页表进行统一管理。反之，若待创建的目标页表不属于重要页表，则按照现有技术，在内核中创建目标页表。具体的，如图10所示，该方法可包括：

[0208] S301、代理模块检测到页表创建命令。

[0209] 其中，页表创建命令用于创建目标页表。

[0210] S302、代理模块判断目标页表是否属于重要页表。若是，则执行如下步骤S303，若否，则执行步骤S304。

[0211] S303、代理模块向控制模块发送页表创建命令。

[0212] S304、代理模块创建目标页表。

[0213] S305、控制模块根据页表创建命令，创建目标页表，并将目标页表存储在专用页表池中。

[0214] 场景三：重映射页表场景

[0215] 如图11所示，该方法可包括：

[0216] S401、代理模块检测到页表重映射命令。

[0217] 其中，页表重映射命令用于重映射目标页表。

[0218] S402、代理模块判断目标页表是否属于重要页表。若是，则执行如下步骤S403，若否，则执行步骤S404。

[0219] S403、代理模块向控制模块发送页表重映射命令。

[0220] S404、代理模块执行页表重映射。

[0221] S405、控制模块根据页表重映射命令，执行页表重映射。

[0222] 上文介绍了开关子模块的工作原理，如下结合举例来说明开关子模块、监测子模块之间配合，实现驱动代码段保护的流程。如图12所示，该方法包括：

[0223] S501、开关子模块检测到插入ko模块的操作。

[0224] S502、开关子模块对ko模块的签名进行校验。

[0225] 示例性的，安全启动开启，开关子模块检测到插入ko模块的操作之后，对ko模块进行签名校验，若校验未通过，则ko模块加载失败，若校验通过，则开关子模块执行步骤S503。

[0226] S503、若校验通过，则代理模块的开关子模块向控制模块的开关子模块发送监测功能关闭命令。

[0227] S504、控制模块的开关子模块向监测子模块发送监测功能关闭命令，用于关闭检测功能。

[0228] S505、响应于监测功能关闭命令，控制模块的监测子模块关闭监测功能。

[0229] 示例性的，开关子模块通知监控模块关闭对非内核代码段区域的合法性判断。

[0230] S506、控制模块的监测子模块修改目标页表。

[0231] 可选的，也可以由代理模块的监测子模块修改目标页表，以便能够插入ko模块。

[0232] S507、控制模块的监测子模块向代理模块的开关子模块发送修改响应，用于指示目标页表已修改完成。

[0233] S508、代理模块的开关子模块插入ko模块。

[0234] S509、代理模块的开关子模块向控制模块的开关子模块发送监测功能恢复命令。

[0235] S510、控制模块的开关子模块向控制模块的监测子模块发送监测功能恢复命令，以便恢复监测功能。

[0236] S511、响应于监测功能恢复命令，控制模块的监测子模块恢复监测功能。

[0237] 本申请实施例中，重要页表的定义可参考关键术语定义里的描述。重要页表的范围，在具体实现专利方案时，可灵活进行控制，如只监控内核代码段的篡改，或者数据段页表只监控高层级页表。

[0238] 如上图13所示为本专利在arm架构上一种实现例子。划分一段内存，将其设置为安全世界，内核都没法访问它。其访问必须是提权之后，即获得安全世界的访问权限后，才能访问。

[0239] 代理程序运行在内核(kernel)中，内核运行在EL2层级的CPU权限上。

[0240] 控制程序运行在负责切换安全世界和非安全世界的可信固件(arm trusted firmware，ATF)模块(BIOS固件(主板中固化的软件(但其实更接近硬件，有别于内核软件代码))的一部分)中。

[0241] 该结构方案适合在运行云化虚拟机(GuestOS)的主机(HostOS)上运行，虚拟机的运行不受影响。该专利方案可以通过保护主机内核内存的完整性，来防止攻击者通过虚拟机入侵主机内核，从而获取其他虚拟机中的重要数据。

[0242] 如上图14所示为本专利在arm架构(自上而下)上一种实现例子。

[0243] 代理程序运行在虚拟机操作系统(GuestOS)的VM内核(kernel)中，VM内核运行在EL1层级的CPU权限上。

[0244] 控制程序运行在主机操作系统(HostOS)的内核(kernel)中。

[0245] 该结构方案适合部署在虚拟机中运行，通过主机内核来保障虚拟机VM内核的完整性。防止攻击者入侵VM内核，从而入侵主机，或者进行其他攻击活动。

[0246] 如上图15所示为本专利在x86_64架构上一种实现例子。

[0247] 可将运行在x86_64架构下root mode下ring0特权的的主机(HostOS)kernel移到

non-root mode下ring0特权运行。两个模式下分别有四种权限，root控制non-root。

[0248] 代理程序运行在主机操作系统(HostOS)的内核kernel中，位于ring0的non-root mode。

[0249] 控制程序运行在ring0的root mode中。

[0250] 该结构方案部署在主机上运行，保障主机的内核kernel的完整性。该方案适合不运行虚拟机的场景。

[0251] 该方案下仍然可以运行虚拟机，不过存在虚拟机嵌套，根据芯片具体实现不同，部分芯片可能会造成一些性能损耗。

[0252] 本申请实施例的技术方案，内核特权下无法直接修改内核中关键内存权限，修改需要经过更高特权中的程序判断合法性后完成。基于本申请实施例的技术方案：可以开辟专用页表池rop-pool，并建立页表自身映射，然后往其中存入内核页表等，实现重要页表的统一管理。利用高权限来完成移除rop-pool中自身映射页表的写入权限，实现内核在页表存在映射环路的情况下，无法全部移除其写入权限的难题。控制程序的合法性判断逻辑：禁止关键的数据被篡改。可以修改重要页表，但必须陷入第二权限来修改。i.判读下发的命令是否是要修改内核代码段区域；ii.内核代码段除个别特殊命令(热补丁命令)外的修改皆为非法；iii.非内核代码段(数据段在开源时就已经具有了写入权限，如果攻击者想加入执行权限，那么控制程序会拒绝)根据地址范围受限添加执行权限或者写入权限，使同一片内存区域在运行过程中无法同时具有执行和写入权限；攻击者通过代理程序触发修改页表权限的操作指令，操作指令(携带待修改的代码的地址范围)会下发到控制程序。通过开关子模块支持“矛盾”功能：保护内核代码段无法被写入，又支持热补丁等对内核代码段的修改。KO在开源时已经有执行权限，写入权限(需要向其中写入)；开关关闭，可以写入代码，写入之后，自动去除写入权限。内核特权，视为最高权限，攻击者可以随意修改。内核中无法禁止修改的数据，采用周期性度量保障完整性。

[0253] 采用本申请实施例的技术方案，攻击者在获取root权限后，无法绕过现有的软件防护机制，即无法修改内核代码段映像中的内容，无法执行在数据段中注入的代码，修改系统寄存器会被检测到并产生告警。本专利方案可抵御ret2usr，ret2dir，KSMA和内核代码注入等大部分内核内存漏洞攻击。本专利方案无法阻止的攻击类型主要有ROP/JOP类的攻击。本专利方案中可以通过周期性度量重要数据，如DMA页表映射数据等，来实现“开源页表权限技术无法防御非通过CPU侧入侵的攻击”的功能。本专利提出的是一种主动实时的防御方案，系统正常运行过程中，几乎不会陷入高权限进行页表权限修改等，所以本专利在系统运行时，可以通过控制重要页表范围，度量周期和度量内容，达到很少的性能损耗。

[0254] 本申请另一些实施例提供了一种装置，该装置可以是上述电子设备(比如折叠屏手机)。该装置可以包括：显示屏、存储器和一个或多个处理器。该显示屏、存储器和处理器耦合。该存储器用于存储计算机程序代码，该计算机程序代码包括计算机指令。当处理器执行计算机指令时，电子设备可执行上述方法实施例中手机执行的各个功能或者步骤。该电子设备的结构可以参考图4所示的电子设备。

[0255] 其中，该电子设备的核心结构可以表示为图16所示的结构，该核心结构可包括：处理单元1301、输入单元1302、存储单元1303、显示单元1304。

[0256] 处理单元1301，可包括中央处理器(CPU)、应用处理器(Application Processor，

AP)或通信处理器(Communication Processor,CP)中的至少一个。处理单元1301可执行与用户电子设备的其他元件中的至少一个的控制和/或通信相关的操作或数据处理。具体地,处理单元1301可用于根据一定的触发条件,控制主屏上显示的内容。或者根据预设规则确定屏幕上显示的内容。处理单元1301还用于将输入的指令或数据进行处理,并根据处理后的数据确定显示样式。

[0257] 输入单元1302,用于获取用户输入的指令或数据,并将获取到的指令或数据传输到电子设备的其他模块。具体地说,输入单元1302的输入方式可以包括触摸、手势、接近屏幕等,也可以是语音输入。例如,输入单元可以是电子设备的屏幕,获取用户的输入操作并根据获取到的输入操作生成输入信号,将输入信号传输至处理单元1301。

[0258] 存储单元1303,可包括易失性存储器和/或非易失性存储器。存储单元用于存储用户终端设备的其他模块中的至少一个相关的指令或数据,具体地说,存储单元可记录页表。

[0259] 显示单元1304,可包括例如液晶显示器(LCD)、发光二极管(LED)显示器、有机发光二极管(OLED)显示器、微机电系统(MEMS)显示器或电子纸显示器。用于显示用户可观看的内容(例如,文本、图像、视频、图标、符号等)。

[0260] 可选的,图16所示结构还可通信单元1305,用于支持电子设备与其他电子设备通信。例如,通信单元可经由无线通信或有线通信连接到网络,以与其他个人终端或网络服务器进行通信。无线通信可采用蜂窝通信协议中的至少一个,诸如,长期演进(LTE)、高级长期演进(LTE-A)、码分多址(CDMA)、宽带码分多址(WCDMA)、通用移动通信系统(UMTS)、无线宽带(WiBro)或全球移动通信系统(GSM)。无线通信可包括例如短距通信。短距通信可包括无线保真(Wi-Fi)、蓝牙、近场通信(NFC)、磁条传输(MST)或GNSS中的至少一个。

[0261] 需要说明的是,本申请方法实施例中的各步骤的描述均可援引到装置对应的模块,这里不再赘述。

[0262] 本申请实施例还提供一种芯片系统,如图17所示,该芯片系统包括至少一个处理器1401和至少一个接口电路1402。处理器1401和接口电路1402可通过线路互联。例如,接口电路1402可用于从其它装置(例如电子设备的存储器)接收信号。又例如,接口电路1402可用于向其它装置(例如处理器1401)发送信号。示例性的,接口电路1402可读取存储器中存储的指令,并将该指令发送给处理器1401。当指令被处理器1401执行时,可使得电子设备执行上述实施例中的各个步骤。当然,该芯片系统还可以包含其他分立器件,本申请实施例对此不作具体限定。

[0263] 本申请实施例还提供一种计算机存储介质,该计算机存储介质包括计算机指令,当计算机指令在上述电子设备上运行时,使得该电子设备执行上述方法实施例中手机执行的各个功能或者步骤。

[0264] 本申请实施例还提供一种计算机程序产品,当计算机程序产品在计算机上运行时,使得计算机执行上述方法实施例中手机执行的各个功能或者步骤。

[0265] 通过以上实施方式的描述,所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,仅以上述各功能模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能模块完成,即将装置的内部结构划分成不同的功能模块,以完成以上描述的全部或者部分功能。

[0266] 在本申请所提供的几个实施例中,应该理解到,所揭露的装置和方法,可以通过其

它的方式实现。例如,以上所描述的装置实施例仅是示意性的,例如,模块或单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个装置,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0267] 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一个物理单元或多个物理单元,即可以位于一个地方,或者也可以分布到多个不同地方。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0268] 另外,在本申请各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0269] 集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时,可以存储在一个可读取存储介质中。基于这样的理解,本申请实施例的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该软件产品存储在一个存储介质中,包括若干指令用以使得一个设备(可以是单片机,芯片等)或处理器(processor)执行本申请各个实施例方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(read only memory,ROM)、随机存取存储器(random access memory,RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0270] 以上内容,仅为本申请的具体实施方式,但本申请的保护范围并不局限于此,任何在本申请揭露的技术范围内的变化或替换,都应涵盖在本申请的保护范围之内。因此,本申请的保护范围应以所述权利要求的保护范围为准。

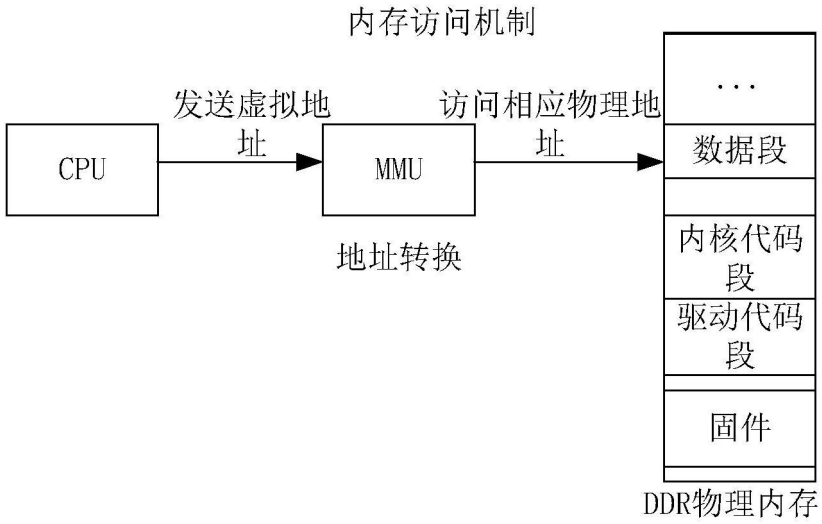

内存访问机制
发送虚拟地址
访问相应物理地址
CPU
MMU
地址转换
...
数据段
内核代码段
驱动代码段
固件
DDR物理内存

图1

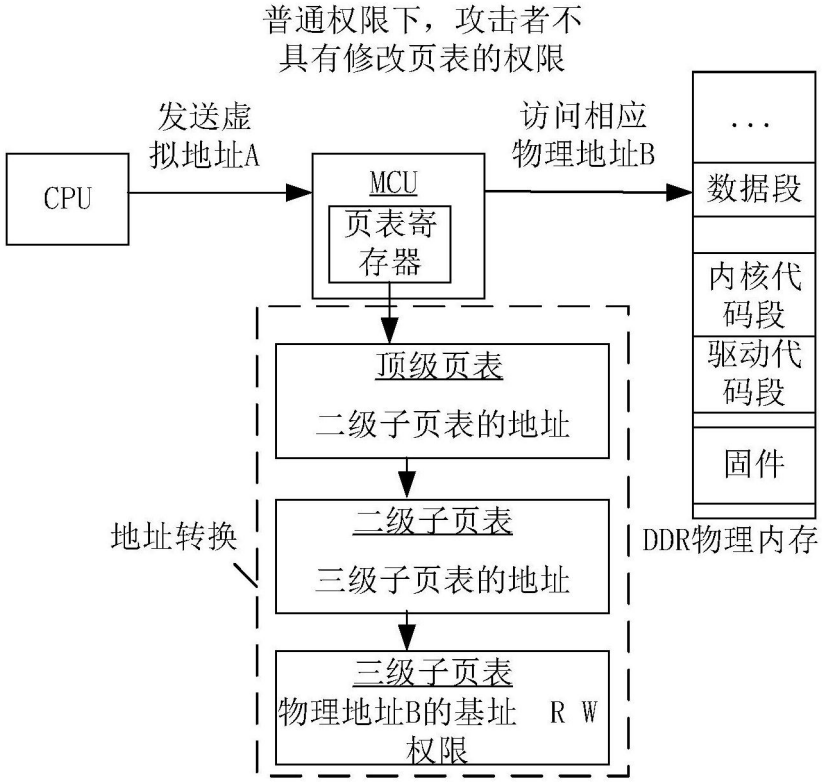

普通权限下，攻击者不具有修改页表的权限
发送虚拟地址A
访问相应物理地址B
CPU
MCU
页表寄存器
...
数据段
内核代码段
驱动代码段
固件
DDR物理内存
顶级页表
二级子页表的地址
二级子页表
三级子页表的地址
三级子页表
物理地址B的基址 R W
权限
地址转换

图2A

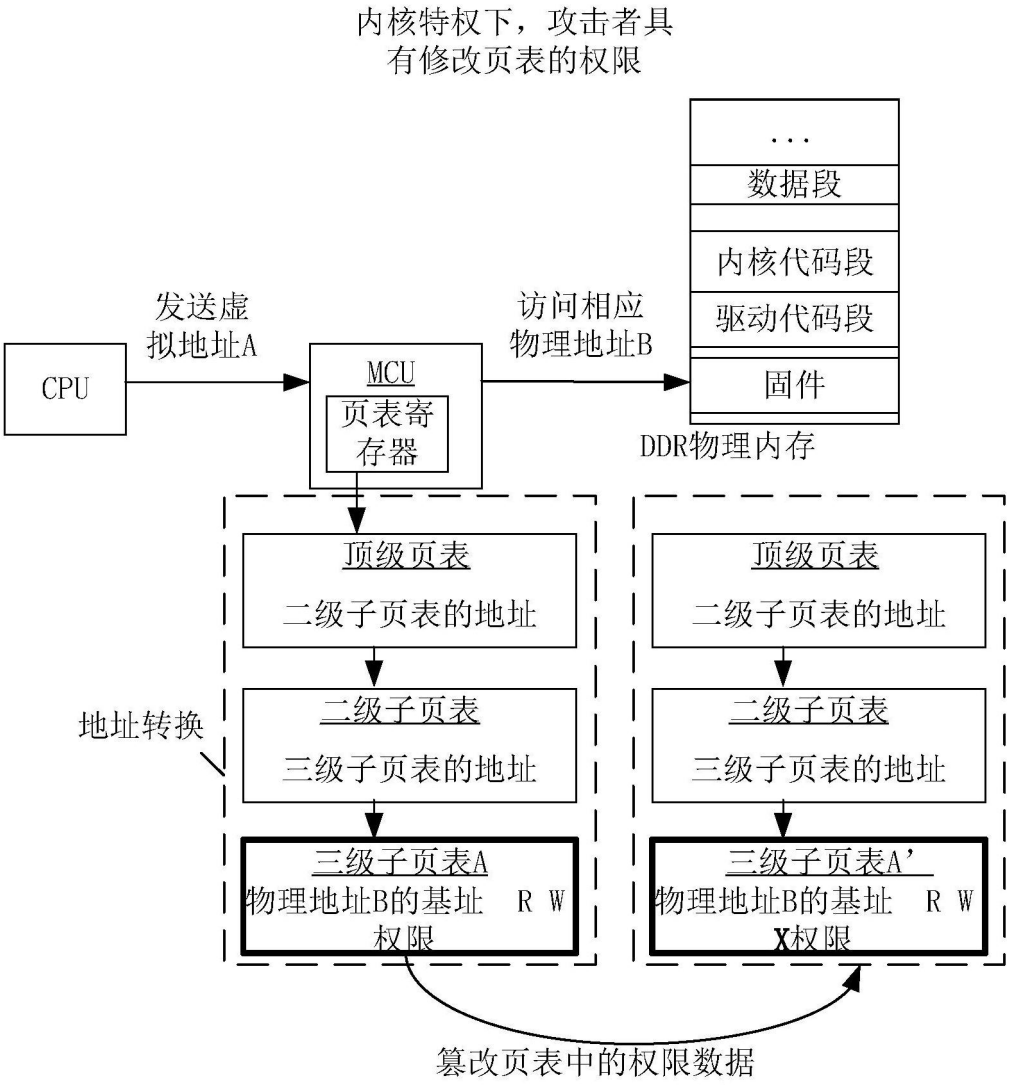

内核特权下，攻击者具
有修改页表的权限
...
数据段
内核代码段
驱动代码段
固件
DDR物理内存
发送虚
拟地址A
访问相应
物理地址B
CPU
MCU
页表寄
存器
顶级页表
二级子页表的地址
二级子页表
三级子页表的地址
三级子页表A
物理地址B的基址 R W
权限
顶级页表
二级子页表的地址
二级子页表
三级子页表的地址
三级子页表A'
物理地址B的基址 R W
X权限
地址转换
篡改页表中的权限数据

图2B

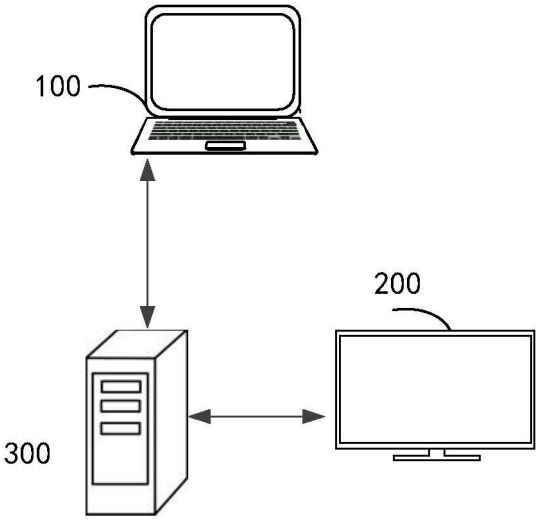


图3

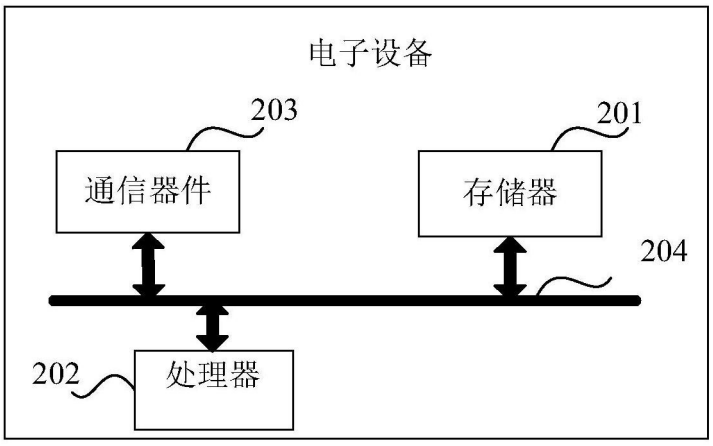


图4

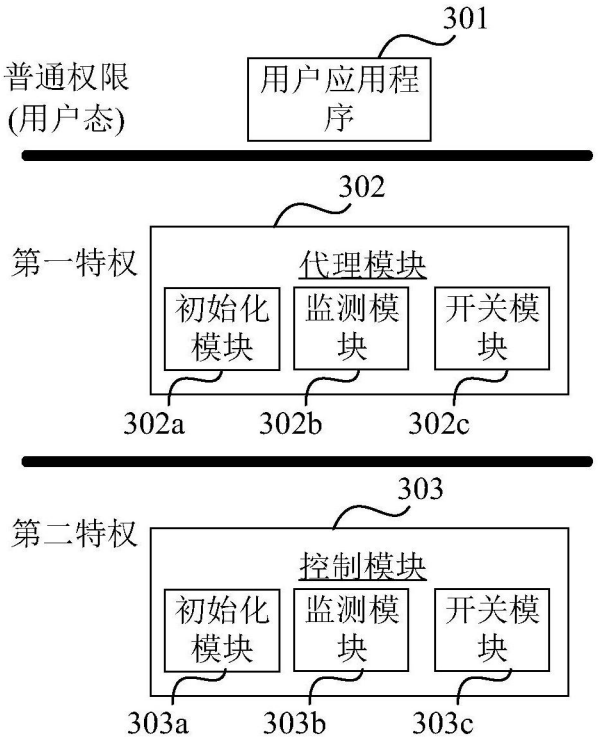

301
普通权限
(用户态)
用户应用程序
302
第一特权
代理模块
初始化模块
监测模块
开关模块
302a
302b
302c
303
第二特权
控制模块
初始化模块
监测模块
开关模块
303a
303b
303c
处理器上运行的软件

图5

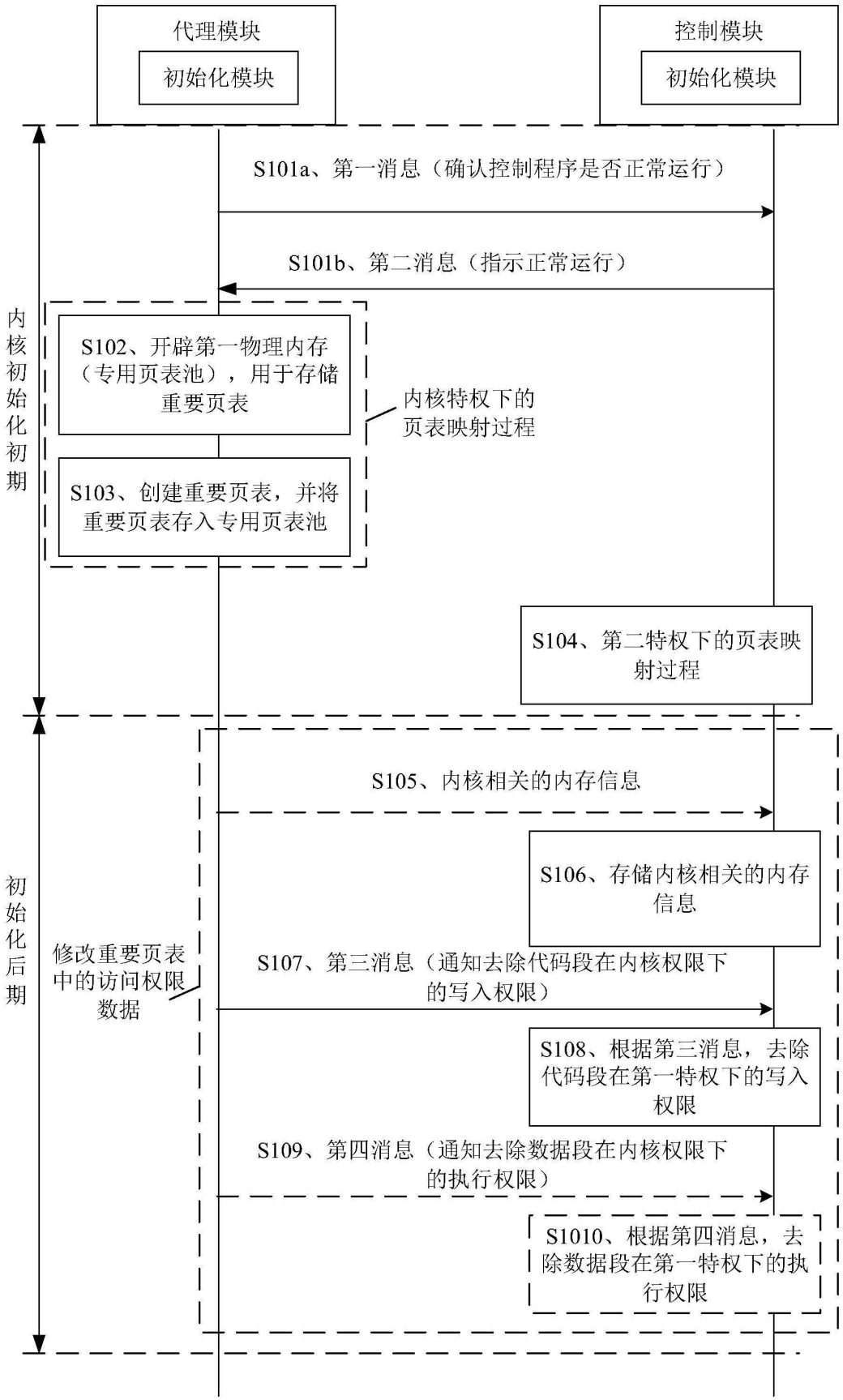
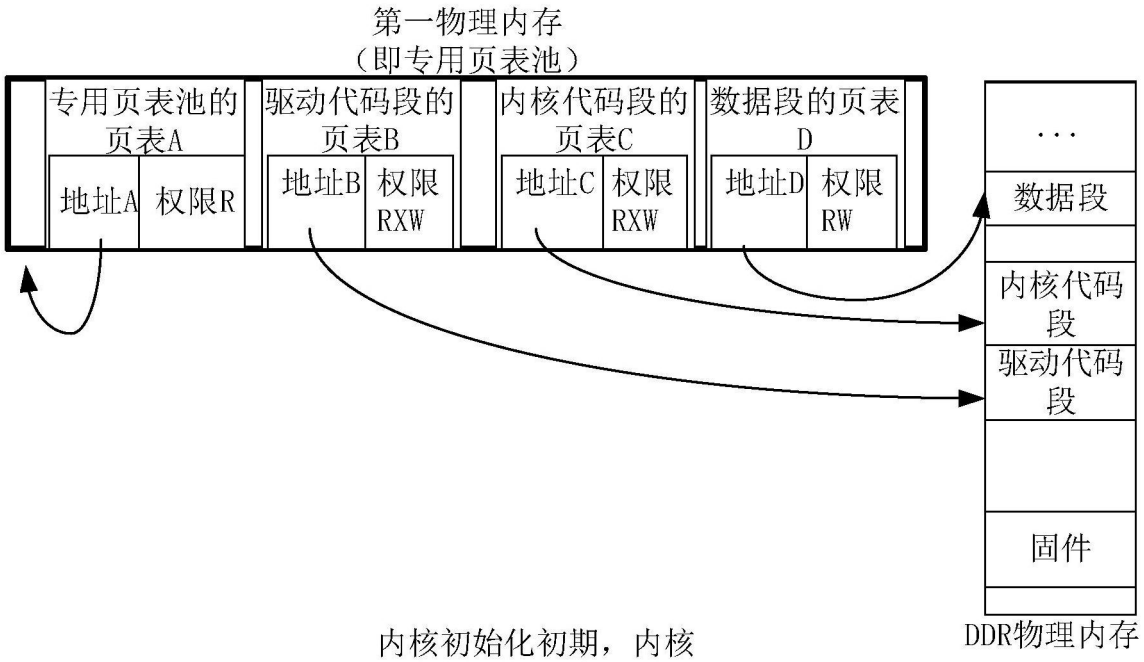
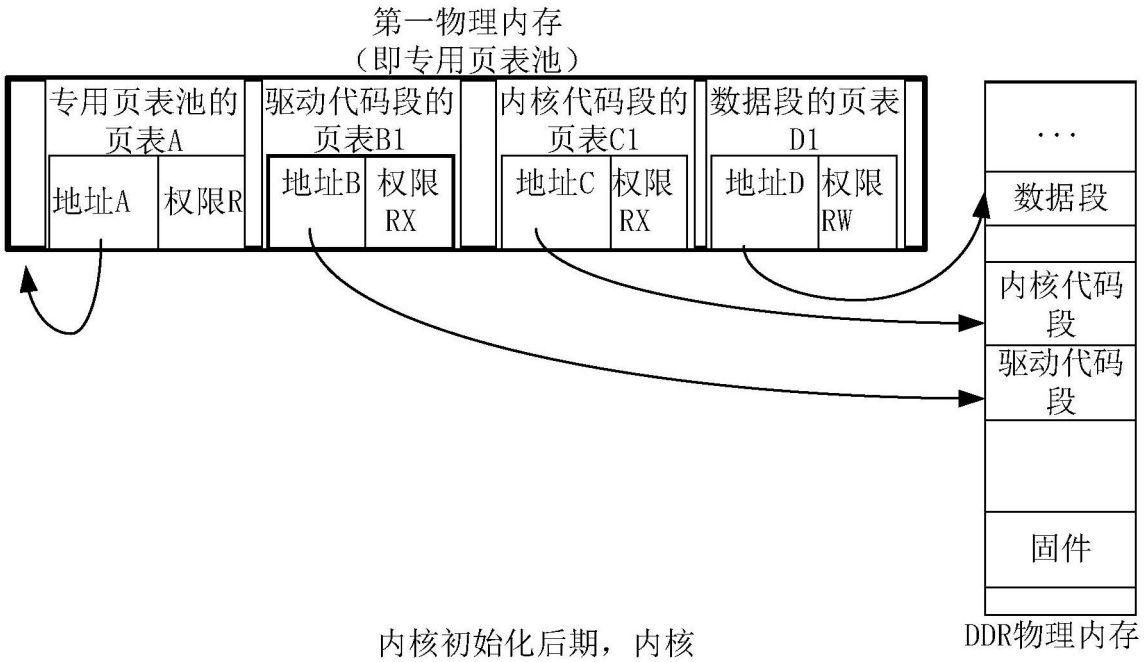

代理模块
初始化模块
控制模块
初始化模块
S101a、第一消息（确认控制程序是否正常运行）
S101b、第二消息（指示正常运行）
内核初始化初期
S102、开辟第一物理内存（专用页表池），用于存储重要页表
内核特权下的页表映射过程
S103、创建重要页表，并将重要页表存入专用页表池
S104、第二特权下的页表映射过程
初始化后期
S105、内核相关的内存信息
S106、存储内核相关的内存信息
修改重要页表中的访问权限数据
S107、第三消息（通知去除代码段在内核权限下的写入权限）
S108、根据第三消息，去除代码段在第一特权下的写入权限
S109、第四消息（通知去除数据段在内核权限下的执行权限）
S1010、根据第四消息，去除数据段在第一特权下的执行权限

图6



内核初始化初期，内核权限下的页表映射

图7A



内核初始化后期，内核权限下的页表映射

图7B

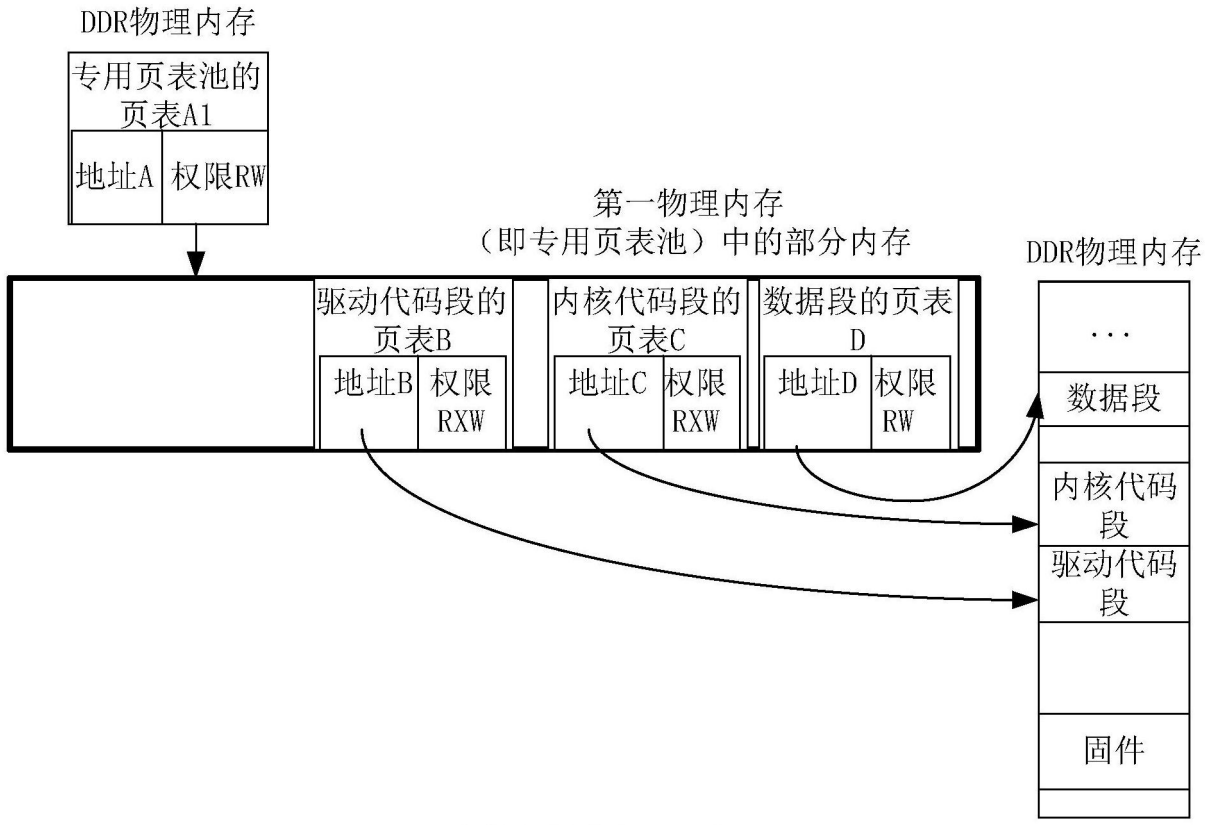

DDR物理内存
专用页表池的页表A1
地址A
权限RW
第一物理内存
（即专用页表池）中的部分内存
驱动代码段的页表B
地址B
权限RXW
内核代码段的页表C
地址C
权限RXW
数据段的页表D
地址D
权限RW
DDR物理内存
...
数据段
内核代码段
驱动代码段
固件
内核初始化初期，第二权限下的页表映射

图8A

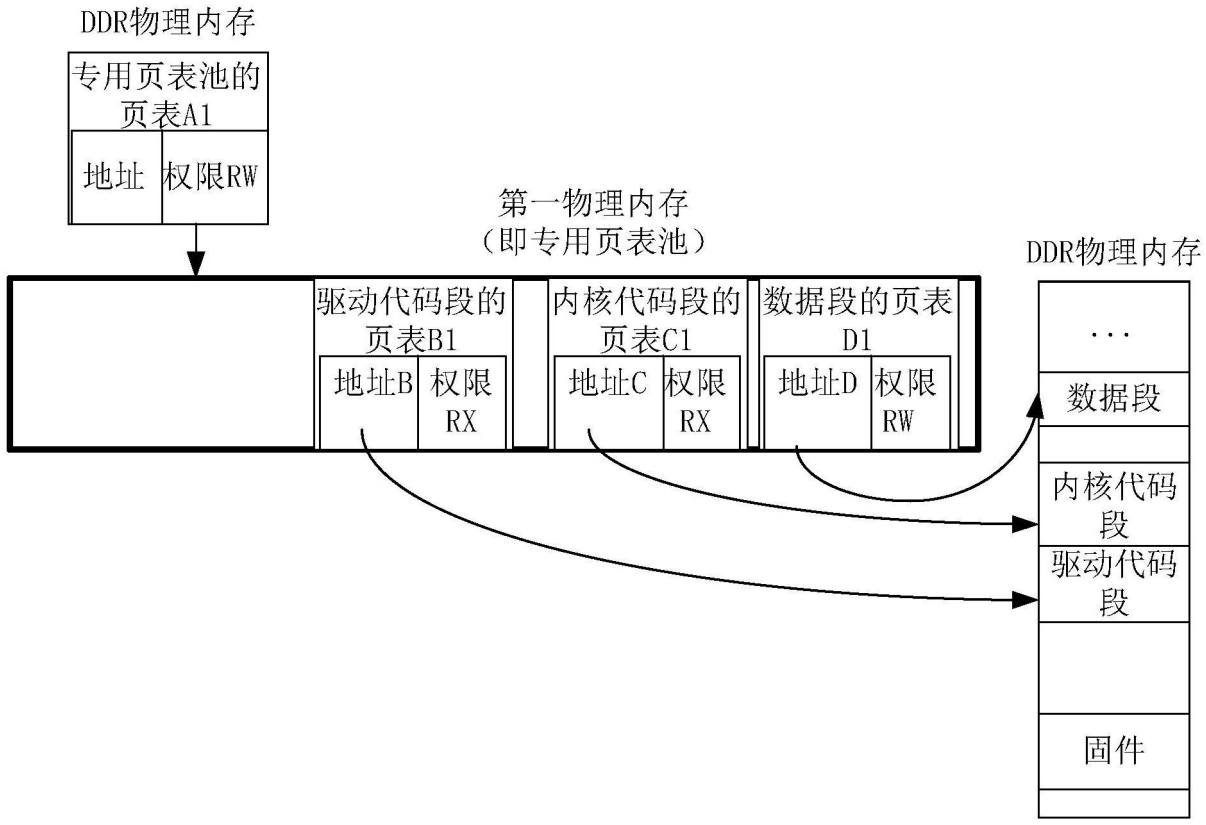

DDR物理内存
专用页表池的
页表A1
地址
权限RW
第一物理内存
（即专用页表池）
驱动代码段的
页表B1
地址B
权限
RX
内核代码段的
页表C1
地址C
权限
RX
数据段的页表
D1
地址D
权限
RW
DDR物理内存
...
数据段
内核代码
段
驱动代码
段
固件
内核初始化后期，第二
权限下的页表映射

图8B

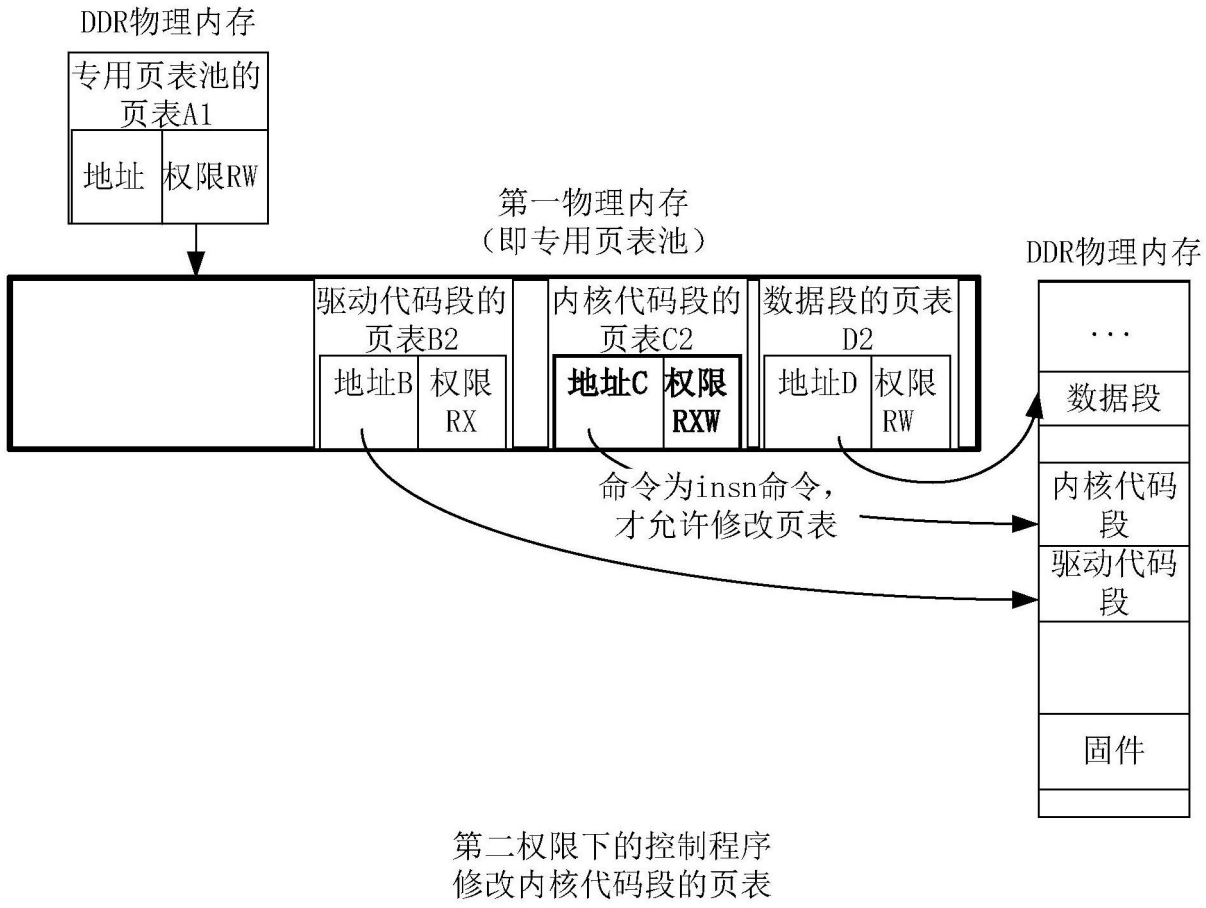

DDR物理内存
专用页表池的
页表A1
地址
权限RW
第一物理内存
（即专用页表池）
驱动代码段的
页表B2
地址B
权限
RX
内核代码段的
页表C2
地址C
权限
RXW
数据段的页表
D2
地址D
权限
RW
DDR物理内存
...
数据段
内核代码
段
驱动代码
段
固件
命令为insn命令，
才允许修改页表
第二权限下的控制程序
修改内核代码段的页表

图8C

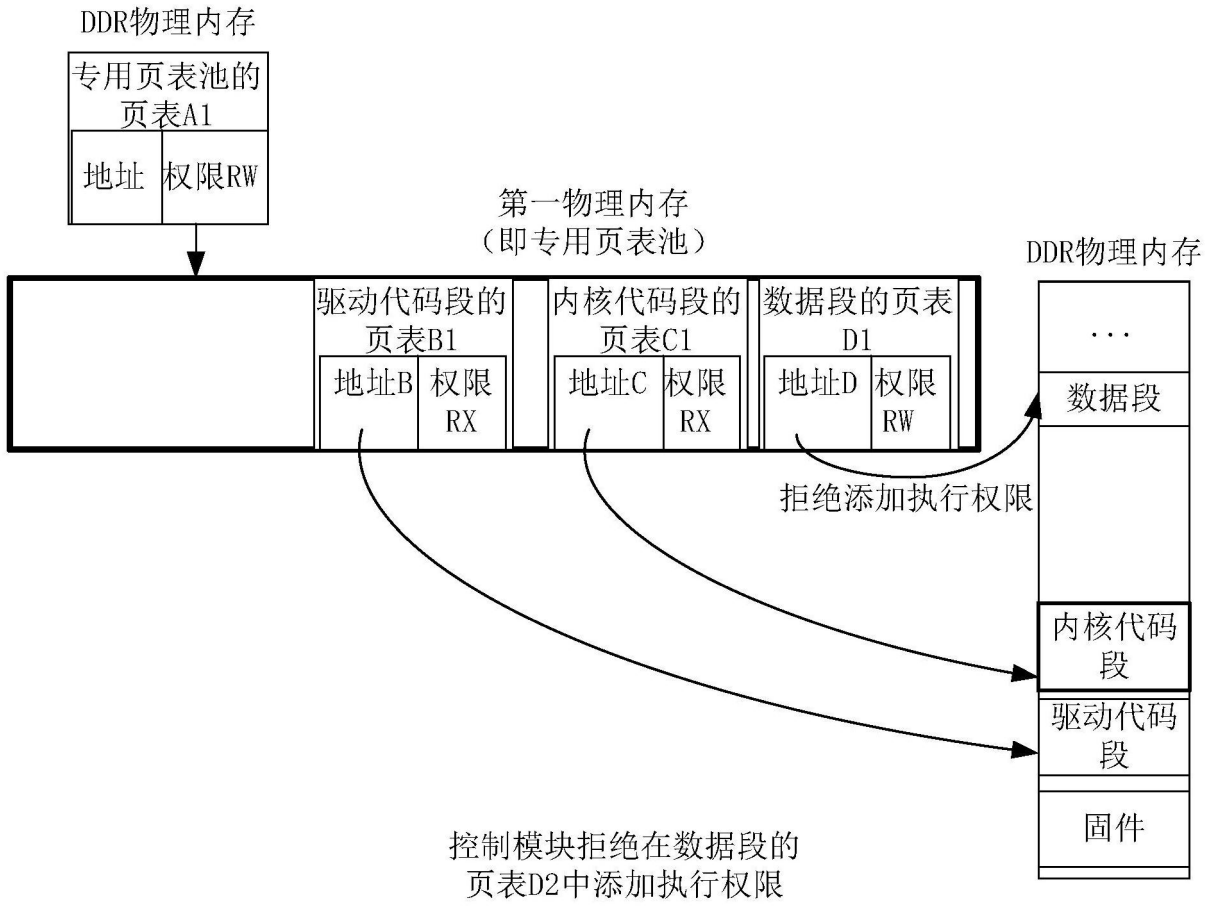

DDR物理内存
专用页表池的
页表A1
地址
权限RW
第一物理内存
（即专用页表池）
驱动代码段的
页表B1
地址B
权限
RX
内核代码段的
页表C1
地址C
权限
RX
数据段的页表
D1
地址D
权限
RW
DDR物理内存
...
数据段
拒绝添加执行权限
内核代码
段
驱动代码
段
固件
控制模块拒绝在数据段的
页表D2中添加执行权限

图8D

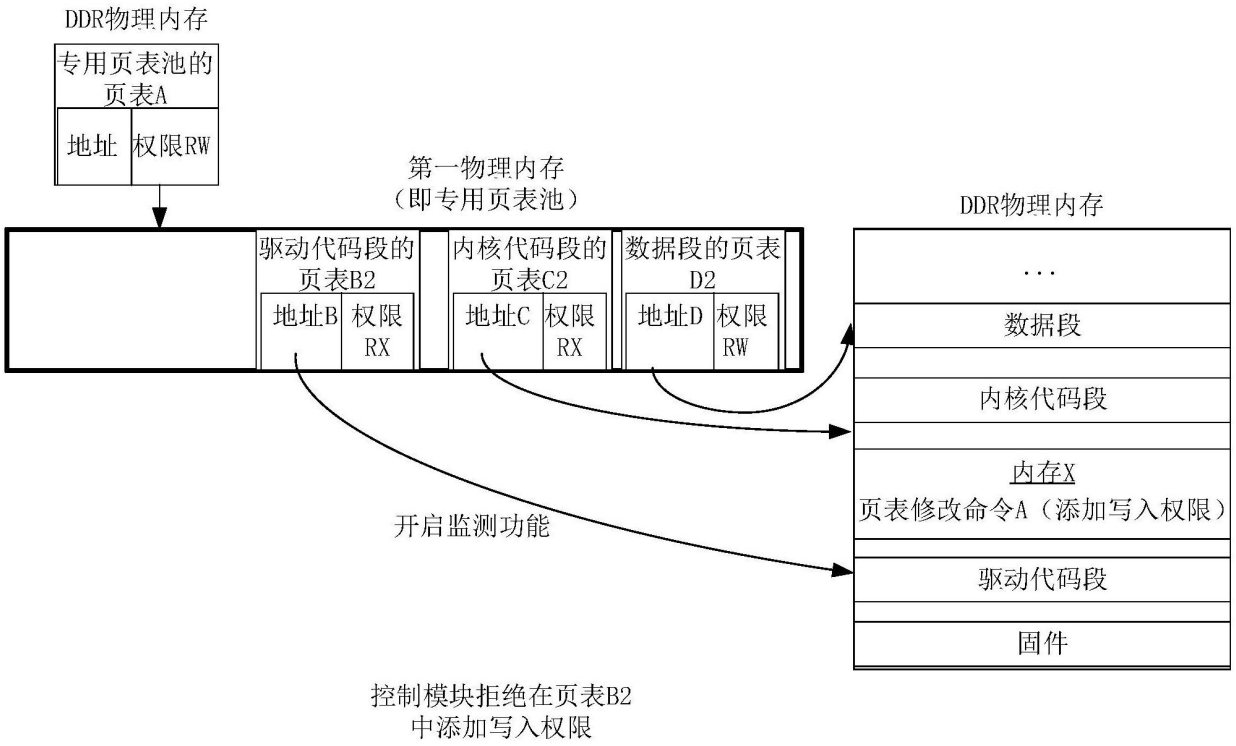

DDR物理内存
专用页表池的页表A
地址
权限RW
第一物理内存
（即专用页表池）
驱动代码段的页表B2
地址B
权限RX
内核代码段的页表C2
地址C
权限RX
数据段的页表D2
地址D
权限RW
DDR物理内存
...
数据段
内核代码段
内存X
页表修改命令A（添加写入权限）
驱动代码段
固件
开启监测功能
控制模块拒绝在页表B2中添加写入权限

图8E

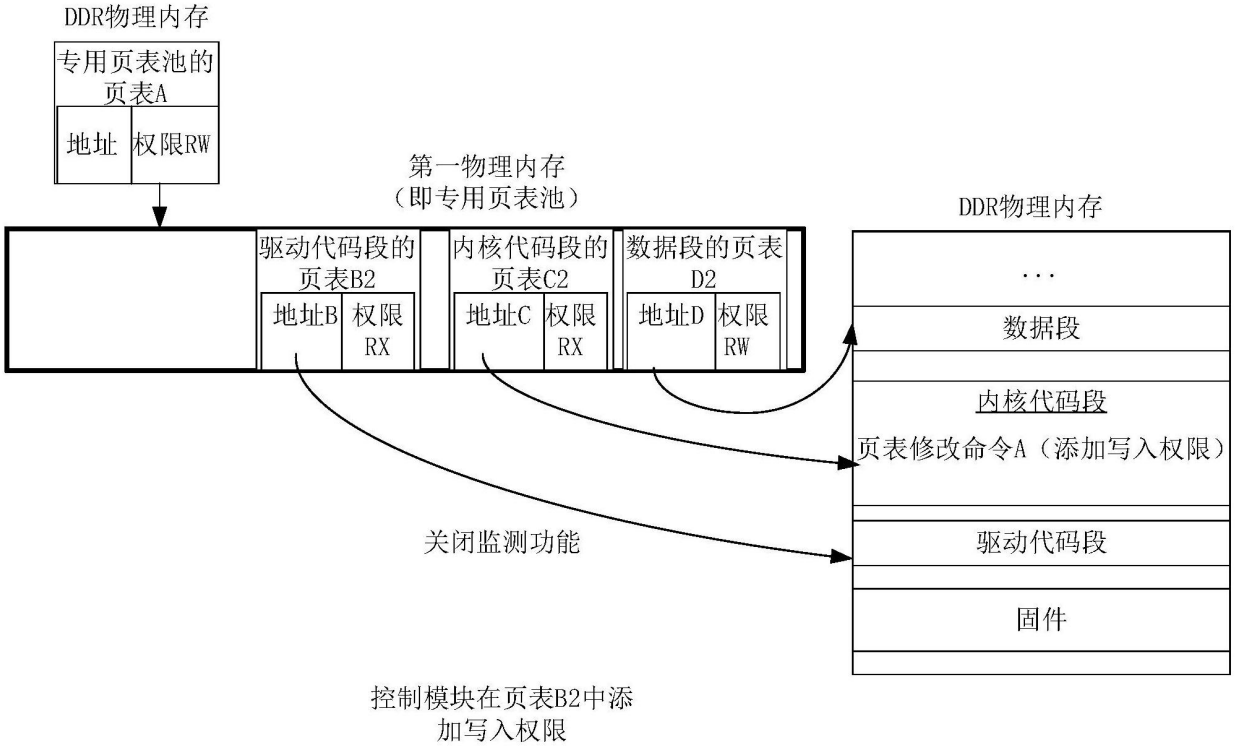

DDR物理内存
专用页表池的页表A
地址
权限RW
第一物理内存
（即专用页表池）
驱动代码段的页表B2
地址B
权限RX
内核代码段的页表C2
地址C
权限RX
数据段的页表D2
地址D
权限RW
DDR物理内存
...
数据段
内核代码段
页表修改命令A（添加写入权限）
驱动代码段
固件
关闭监测功能
控制模块在页表B2中添加写入权限

图8F

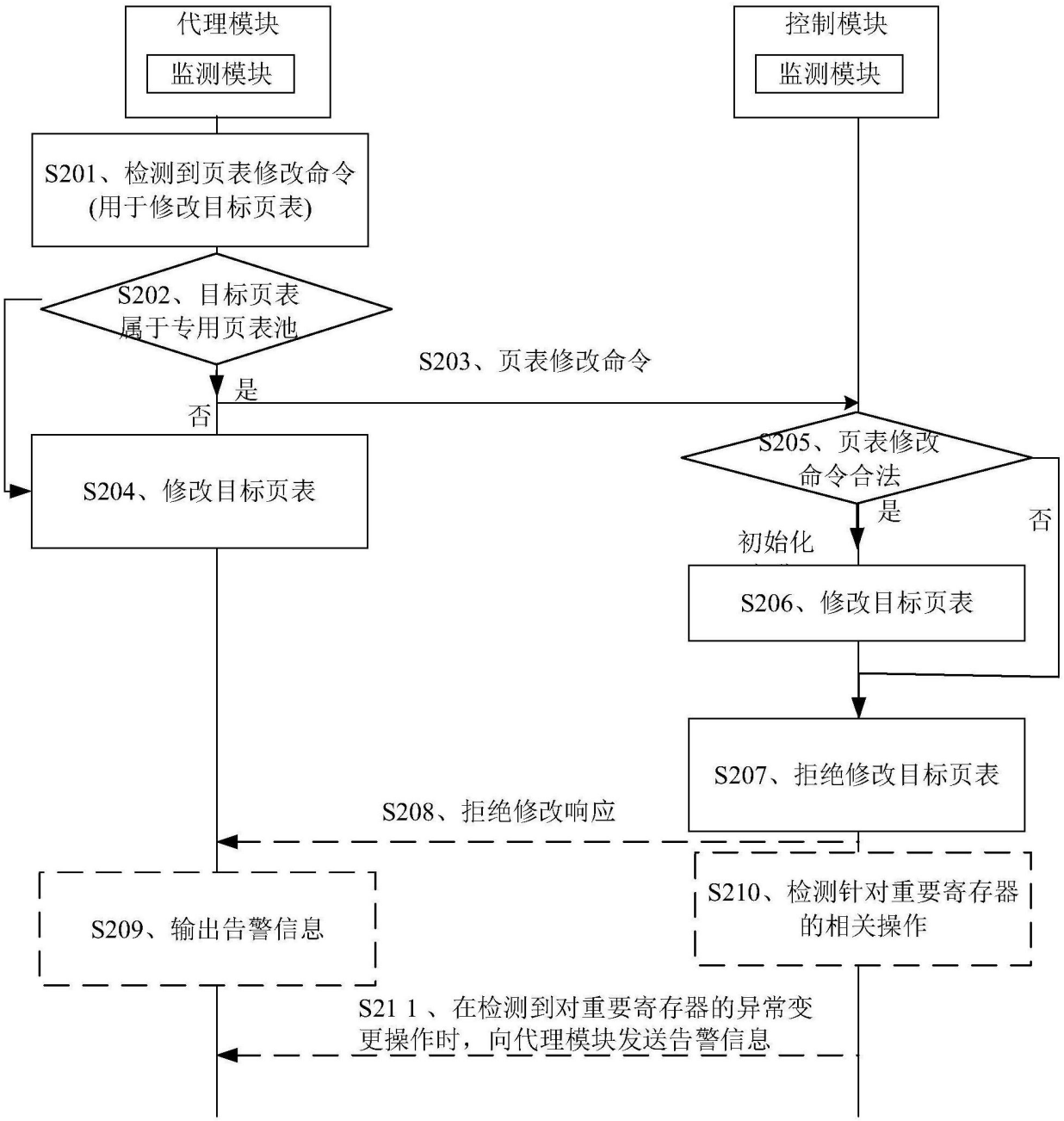

代理模块
监测模块
控制模块
监测模块
S201、检测到页表修改命令
(用于修改目标页表)
S202、目标页表
属于专用页表池
S203、页表修改命令
是
否
S204、修改目标页表
S205、页表修改
命令合法
是
否
初始化
S206、修改目标页表
S207、拒绝修改目标页表
S208、拒绝修改响应
S209、输出告警信息
S210、检测针对重要寄存器
的相关操作
S21 1 、在检测到对重要寄存器的异常变
更操作时，向代理模块发送告警信息

图9

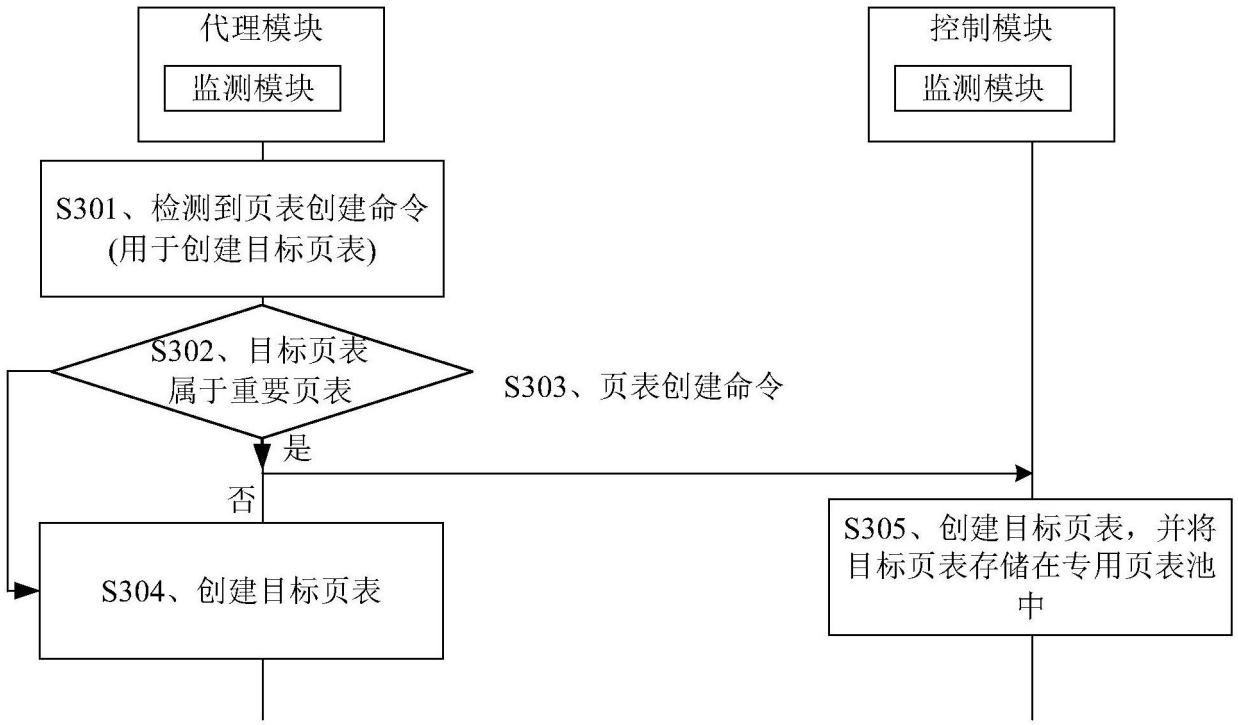

代理模块
监测模块
控制模块
监测模块
S301、检测到页表创建命令(用于创建目标页表)
S302、目标页表属于重要页表
S303、页表创建命令
是
否
S304、创建目标页表
S305、创建目标页表，并将目标页表存储在专用页表池中

图10

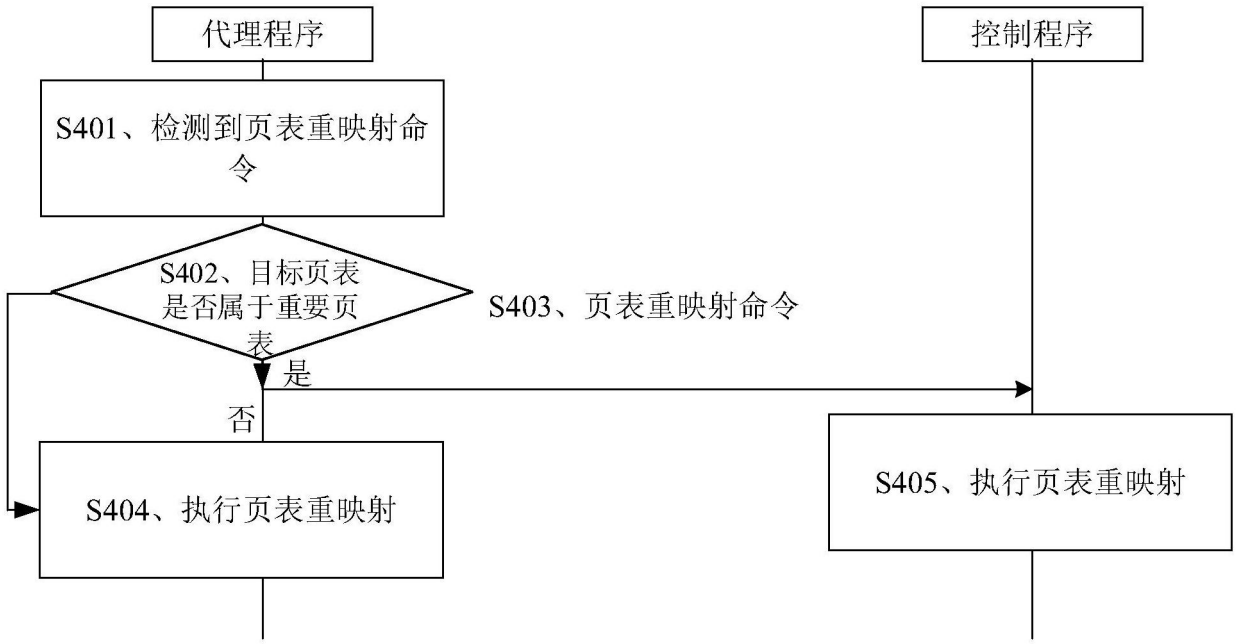

代理程序
控制程序
S401、检测到页表重映射命令
S402、目标页表是否属于重要页表
S403、页表重映射命令
是
否
S404、执行页表重映射
S405、执行页表重映射

图11

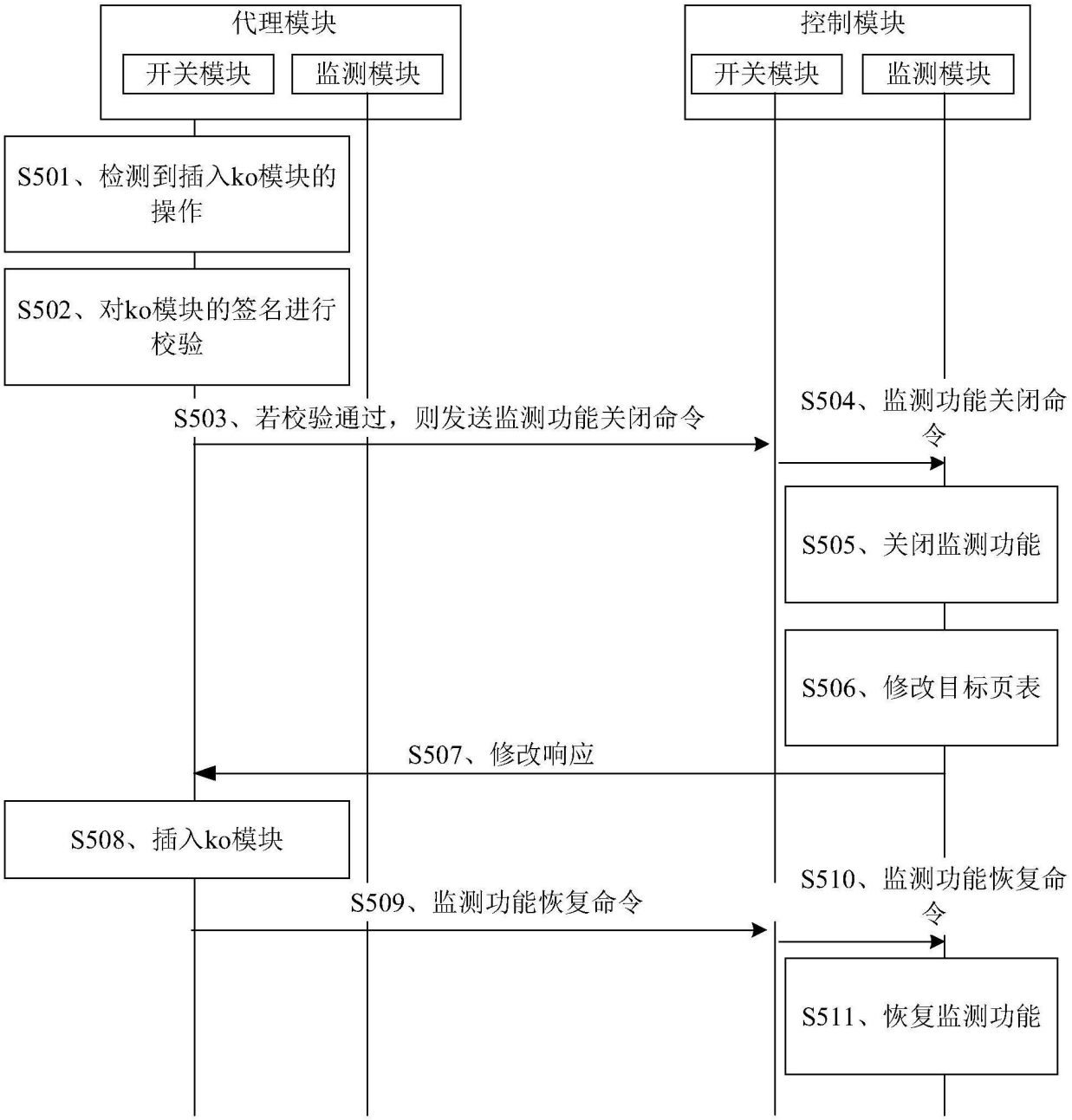

代理模块
开关模块
监测模块
控制模块
开关模块
监测模块
S501、检测到插入ko模块的操作
S502、对ko模块的签名进行校验
S503、若校验通过，则发送监测功能关闭命令
S504、监测功能关闭命令
S505、关闭监测功能
S506、修改目标页表
S507、修改响应
S508、插入ko模块
S509、监测功能恢复命令
S510、监测功能恢复命令
S511、恢复监测功能

图12

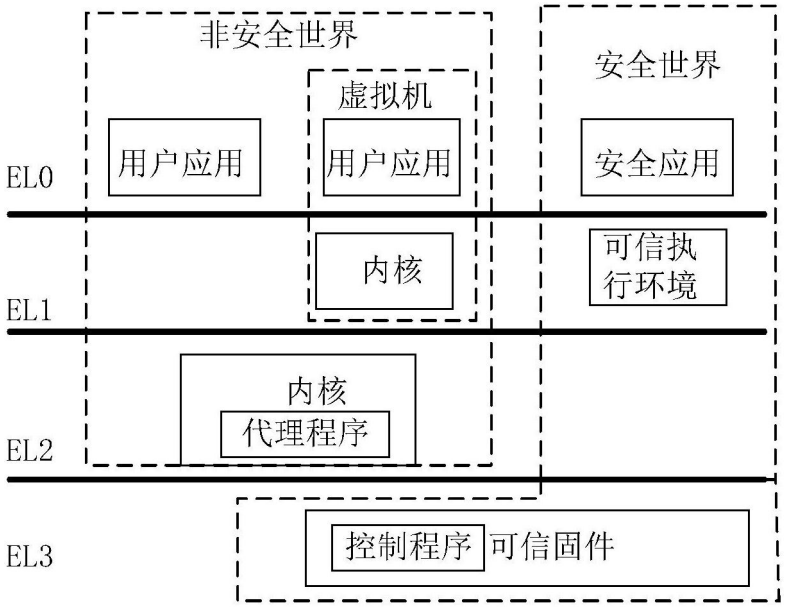

非安全世界
虚拟机
用户应用
用户应用
安全世界
安全应用
EL0
内核
可信执行环境
EL1
内核
代理程序
EL2
控制程序
可信固件
EL3

图13

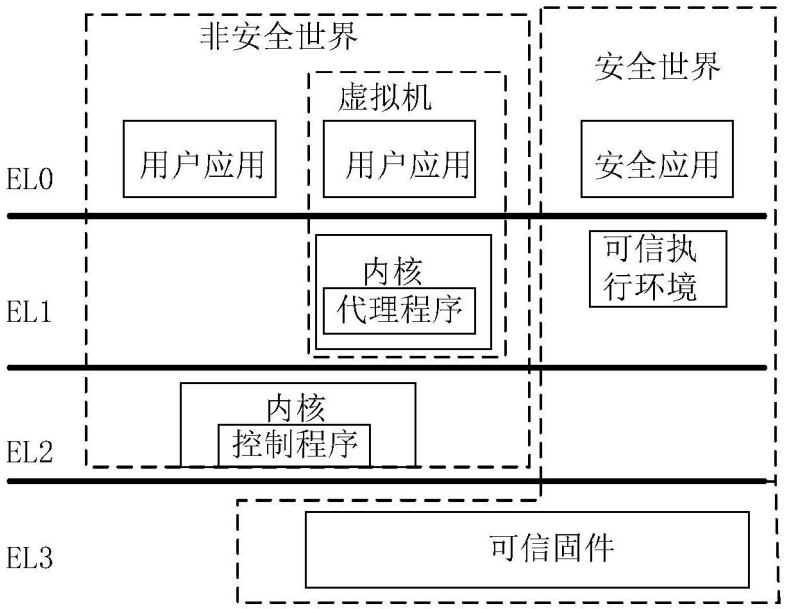

非安全世界
虚拟机
用户应用
用户应用
安全世界
安全应用
EL0
内核
代理程序
可信执行环境
EL1
内核
控制程序
EL2
可信固件
EL3

图14

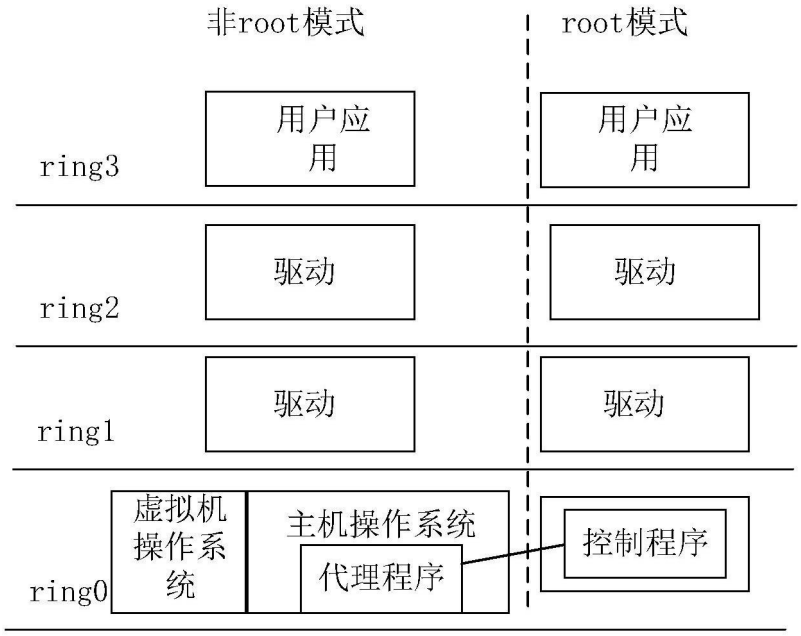

非root模式
root模式
ring3
用户应用
用户应用
ring2
驱动
驱动
ring1
驱动
驱动
ring0
虚拟机操作系统
主机操作系统
代理程序
控制程序

图15

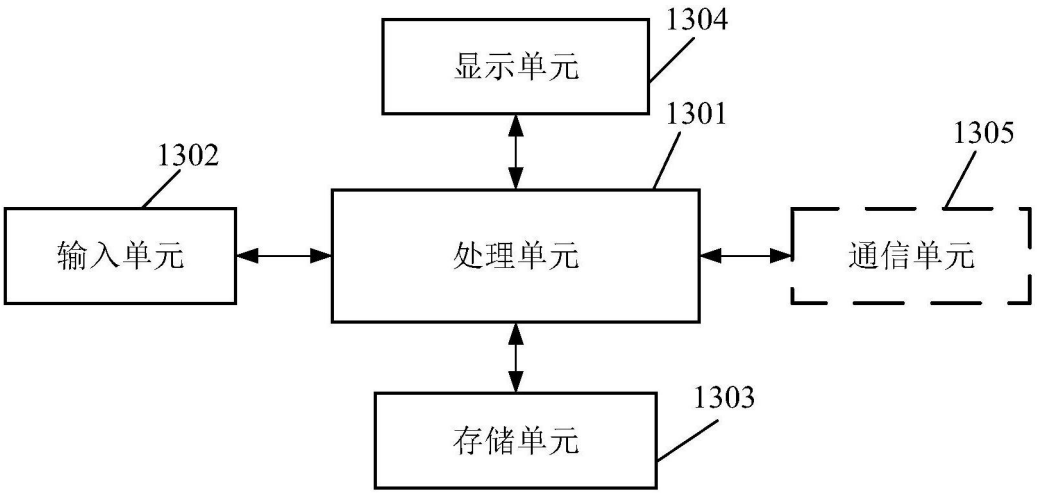

显示单元
1304
1301
1305
1302
输入单元
处理单元
通信单元
存储单元
1303

图16

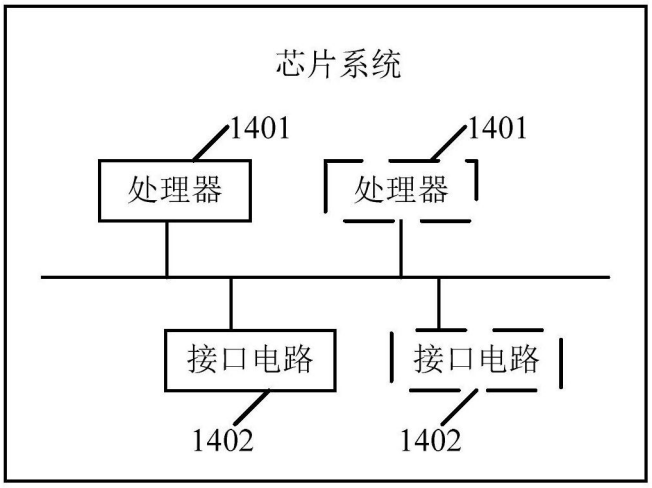


图17