(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 116484381 A
(43)申请公布日 2023.07.25

(21)申请号 202310366631.4

(22)申请日 2023.04.07

(71)申请人 北京奕斯伟计算技术股份有限公司
地址 100176 北京市大兴区北京经济技术开发区科创十街18号院3号楼1层101室

(72)发明人 徐锦涛

(74)专利代理机构 北京天昊联合知识产权代理有限公司 11112
专利代理师 彭瑞欣 吴俣

(51)Int.Cl.
G06F 21/57(2013.01)
G06F 21/52(2013.01)

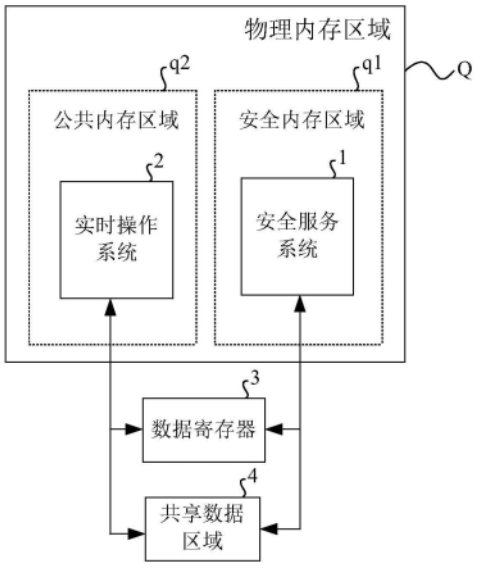
权利要求书2页 说明书9页 附图7页

(54)发明名称

数据处理方法及其系统、电子设备

(57)摘要

本公开提供了一种数据处理方法，应用于安全服务系统侧，包括：响应于第一指令，运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数，所述待处理参数为实时操作系统所运行任务中的参数；运行所述目标安全服务函数对所述待处理参数进行处理；根据所述目标安全服务函数的运行处理结果将结果信息写入至所述数据寄存器，以供所述实时操作系统从所述数据寄存器中读取结果信息。在本公开实施例中，RTOS所运行任务中的参数的处理是在安全内存区域内进行，可以有效保证RTOS所运行任务中的参数的处理过程中不会受到恶意程序的攻击，从而能够保证数据处理过程的安全性。



CN 116484381 A

1.一种数据处理方法,应用于安全服务系统侧,其特征在于,包括:

响应于第一指令,运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数,所述待处理参数为实时操作系统所运行任务中的参数;

运行所述目标安全服务函数对所述待处理参数进行处理;

根据所述目标安全服务函数的运行处理结果将结果信息写入至所述数据寄存器,以供所述实时操作系统从所述数据寄存器中读取结果信息。

2.根据权利要求1所述的数据处理方法,其特征在于,所述安全服务系统包括所述中断异常处理程序和安全服务函数集合;

在根据数据寄存器内的待处理参数确定待调用的目标安全服务函数的步骤之前,还包括:

将安全服务系统所在物理内存区域设置为安全内存区域以及将所述实时操作系统所在物理内存区域设置为公共内存区域,所述安全内存区域的读写权限配置为只读,所述公共内存区域的读写权限配置为可读且可写。

3.根据权利要求2所述的数据处理方法,其特征在于,将安全服务系统所在物理内存区域设置为安全内存区域以及将所述实时操作系统所在物理内存区域设置为公共内存区域的步骤之后,且在运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数的步骤之前,还包括:

响应于第二指令,运行中断异常处理程序将mstatus寄存器内MPP域设置用户模式所代表的值,以及将mepc寄存器内的地址修改为所述实时操作系统的首地址;

运行第三指令以启动所述实时操作系统。

4.根据权利要求1所述的数据处理方法,其特征在于,在根据所述目标安全服务函数的运行处理结果将结果信息写入至所述数据寄存器的步骤之后,还包括:

运行第四指令以控制实时操作系统读取所述数据寄存器内的所述结果信息。

5.根据权利要求1所述的数据处理方法,其特征在于,所述根据所述目标安全服务函数的运行处理结果将结果信息写入至所述数据寄存器的步骤包括:

在所述处理结果为单个数值时,将所述单个数值作为所述结果信息写入至所述数据寄存器;

在所述处理结果为由多个数值所构成的数组时,将所述数组存储至预先设置的共享数据区域,并将所述数组在所述共享数据区域内的首地址作为所述结果信息写入至所述数据寄存器。

6.根据权利要求1至5中任一所述的数据处理方法,其特征在于,所述数据寄存器为整数寄存器或浮点寄存器。

7.一种数据处理方法,应用于实时操作系统侧,其特征在于,包括:

将所运行任务中的参数作为待处理参数存储至所述数据寄存器内;

运行第一指令,使得安全服务系统运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数,以及运行所述目标安全服务函数对所述待处理参数进行处理,并根据所述目标安全服务函数的运行处理结果将结果信息写入至所述数据寄存器;

从所述数据寄存器中读取所述结果信息,并根据所述结果信息得到所述运行处理结

果。

8.根据权利要求7所述的数据处理方法，其特征在于，在将所运行任务中的参数作为待处理参数存储至所述数据寄存器内的步骤之前，还包括：

保存所述任务内所述待处理参数的上下文内容。

9.根据权利要求7所述的数据处理方法，其特征在于，所述从所述数据寄存器中读取所述结果信息，并根据所述结果信息得到所述运行处理结果的步骤包括：

在所述结果信息为单个数值时，将所述单个数值作为所述运行处理结果；

在所述结果信息为所述共享数据区域内的地址时，根据所述地址从所述共享数据区域内获取由多个数值所构成的数组，并将所述数组作为所述运行处理结果。

10.一种安全服务系统，其特征在于，包括：

第一运行模块，配置为响应于第一指令，运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数，所述待处理参数为实时操作系统所运行任务中的参数；

第二运行模块，配置为运行所述目标安全服务函数对所述待处理参数进行处理；

第一写入模块，配置为根据所述目标安全服务函数的运行处理结果将结果信息写入至所述数据寄存器，以供所述实时操作系统从所述数据寄存器中读取结果信息。

11.一种实时操作系统，其特征在于，包括：

第二写入模块，配置为将所运行任务中的参数作为待处理参数写入至所述数据寄存器内；

指令运行模块，配置为运行第一指令，使得安全服务系统运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数，以及运行所述目标安全服务函数对所述待处理参数进行处理，并根据所述目标安全服务函数的运行处理结果将结果信息写入至所述数据寄存器；

读取模块，配置为从所述数据寄存器中读取所述结果信息，并根据所述结果信息得到所述运行处理结果。

12.一种数据处理系统，其特征在于，包括：如权利要求10中所述安全服务系统和如权利要求11中所述实时操作系统。

13.一种电子设备，其特征在于，包括：

处理器；

存储器，用于存储一个或多个程序；

当所述一个或多个程序被所述处理器执行，使得所述处理器实现如权利要求1至6中任一所述数据处理方法中的步骤，和/或权利要求7至9中任一所述数据处理方法中的步骤。

数据处理方法及其系统、电子设备

技术领域

[0001] 本发明涉及计算机领域，特别涉及一种数据处理方法及其系统、电子设备。

背景技术

[0002] CPU芯片被广泛应用在电力、金融、通信等领域，在基于信息化、网络化时代背景下，不仅为促进社会生产、提高社会生产效率注入了动力，同时还为人们的工作与学习提供了便利。基于开放性和共享性特点，使得计算机安全问题随之凸显，并严重威胁到了信息的安全。

[0003] 在实际应用计算机过程中，一些恶意程序会通过CPU芯片的漏洞入侵来获取相关的安全信息。具体地，在数据处理过程中，恶意程序会篡改处理数据的程序(具体为服务函数)以及处理过程中的数据，这会使得计算机信息的安全受到威胁。

发明内容

[0004] 第一方面，本公开实施例提供了一种数据处理方法，应用于安全服务系统侧，其特征在于，包括：响应于第一指令，运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数，待处理参数为实时操作系统所运行任务中的参数；运行目标安全服务函数对待处理参数进行处理；根据目标安全服务函数的运行处理结果将结果信息写入至数据寄存器，以供实时操作系统从数据寄存器中读取结果信息。

[0005] 在一些实施例中，所述安全服务系统包括所述中断异常处理程序和安全服务函数集合；在根据数据寄存器内的待处理参数确定待调用的目标安全服务函数的步骤之前，还包括：将安全服务系统所在物理内存区域设置为安全内存区域以及将所述实时操作系统所在物理内存区域设置为公共内存区域，所述安全内存区域的读写权限配置为只读，所述公共内存区域的读写权限配置为可读且可写。

[0006] 在一些实施例中，将安全服务系统所在物理内存区域设置为安全内存区域以及将所述实时操作系统所在物理内存区域设置为公共内存区域的步骤之后，且在运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数的步骤之前，还包括：响应于第二指令，运行中断异常处理程序将mstatus寄存器内MPP域设置用户模式所代表的值，以及将mepc寄存器内的地址修改为所述实时操作系统的首地址；运行第三指令以启动所述实时操作系统。

[0007] 在一些实施例中，在根据所述目标安全服务函数的运行处理结果将结果信息写入至所述数据寄存器的步骤之后，还包括：运行第四指令以控制实时操作系统读取所述数据寄存器内的所述结果信息。

[0008] 在一些实施例中，所述根据所述目标安全服务函数的运行处理结果将结果信息写入至所述数据寄存器的步骤包括：在所述处理结果为单个数值时，将所述单个数值作为所述结果信息写入至所述数据寄存器；在所述处理结果为由多个数值所构成的数组时，将所述数组存储至预先设置的共享数据区域，并将所述数组在所述共享数据区域内的首地址作

为所述结果信息写入至所述数据寄存器。

[0009] 在一些实施例中，所述数据寄存器为整数寄存器或浮点寄存器。

[0010] 第二方面，本公开实施例还提供了一种数据处理方法，应用于实时操作系统侧，包括：将所运行任务中的参数作为待处理参数存储至所述数据寄存器内；运行第一指令，使得安全服务系统运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数，以及运行所述目标安全服务函数对所述待处理参数进行处理，并根据所述目标安全服务函数的运行处理结果将结果信息写入至所述数据寄存器；从所述数据寄存器中读取所述结果信息，并根据所述结果信息得到所述运行处理结果。

[0011] 在一些实施例中，在将所运行任务中的参数作为待处理参数存储至所述数据寄存器内的步骤之前，还包括：保存所述任务内所述待处理参数的上下文内容。

[0012] 在一些实施例中，所述从所述数据寄存器中读取所述结果信息，并根据所述结果信息得到所述运行处理结果的步骤包括：在所述结果信息为单个数值时，将所述单个数值作为所述运行处理结果；在所述结果信息为所述共享数据区域内的地址时，根据所述地址从所述共享数据区域内获取由多个数值所构成的数组，并将所述数组作为所述运行处理结果。

[0013] 第三方面，本公开实施例提供了一种安全服务系统包括，包括：第一运行模块，配置为响应于第一指令，运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数，待处理参数为实时操作系统所运行任务中的参数；第二运行模块，配置为运行目标安全服务函数对待处理参数进行处理；第一写入模块，配置为根据目标安全服务函数的运行处理结果将结果信息写入至数据寄存器，以供实时操作系统从数据寄存器中读取结果信息。

[0014] 第四方面，本公开实施例提供了一种实时操作系统，包括：第二写入模块，配置为将所运行任务中的参数作为待处理参数写入至数据寄存器内；指令运行模块，配置为运行第一指令，使得安全服务系统运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数，以及运行目标安全服务函数对待处理参数进行处理，并根据目标安全服务函数的运行处理结果将结果信息写入至数据寄存器；读取模块，配置为从数据寄存器中读取结果信息，并根据结果信息得到运行处理结果。

[0015] 第五方面，本公开实施例提供了一种数据处理系统，包括：如第三方面中的安全服务系统和如第四方面中的实时操作系统。

附图说明

[0016] 图1为本公开实施例中一种物理内存区域分布示意图；

[0017] 图2为本公开实施例中一种安全服务交互图；

[0018] 图3为本公开实施例中应用于安全服务系统的一种数据处理方法的流程图；

[0019] 图4为本公开实施例中应用于安全服务系统的另一种数据处理方法的流程图；

[0020] 图5为本公开实施例中应用于实时操作系统的一种数据处理方法的流程图；

[0021] 图6为本公开实施例中应用于实时操作系统的另一种数据处理方法的流程图；

[0022] 图7为本公开实施例中安全服务系统和实时操作系统实现数据处理方法的一种信令图；

[0023] 图8A为本公开实施例中安全服务系统的一种结构框图；

[0024] 图8B为本公开实施例中实时操作系统的一种结构框图；

[0025] 图9为本公开实施例中数据处理系统的一种结构框图；

[0026] 图10为本公开实施例中提供的电子设备的一种结构框图。

具体实施方式

[0027] 为使本领域的技术人员更好地理解本发明的技术方案，下面结合附图对本发明提供的一种数据处理方法及其系统、安全服务系统、实时操作系统、电子设备和计算机可读介质进行详细描述。

[0028] 本公开的技术方案基于RISC-V的标准安全扩展，提供一种能够有效保证数据安全的数据处理方案。

[0029] 图1为本公开实施例中一种物理内存区域分布示意图，图2为本公开实施例中一种安全服务交互图。如图1和图2所示，该方案通过RISC-V的物理内存保护(Physical Memory Protection，简称PMP)安全扩展，实现将安全服务(Security Service)系统1所在物理内存区域Q设置为安全内存区域q1，以及将实时操作系统2(Real Time Operating System，简称RTOS)所在物理内存区域Q设置为公共内存区域q2。安全内存区域q1的读写权限配置为只读，也就是说，位于安全内存区域q1中的数据无法被外部篡改；公共内存区域q2的读写权限配置为可读且可写。

[0030] 其中，RTOS2中运行有用户程序，可通过用户程序来运行相应任务。在运行于RTOS2中的用户程序所运行任务需要调用安全服务系统1内的安全服务函数时，可基于RISC-V框架提供的中断异常处理机制将该任务中相应参数传递至位于安全内存区域的目标安全函数服务进行处理，待目标安全服务函数完成对数据的相应处理后，再将相应结果信息反馈给RTOS2，以供RTOS2继续运行任务。在本公开实施例中，用户程序所运行任务中的参数的处理是在安全内存区域内进行，可以有效保证RTOS2所运行任务中的参数的处理过程不会受到恶意程序的攻击，从而能够保证数据处理过程的安全性。后面将结合具体实施例作详细描述。

[0031] 图3为本公开实施例中应用于安全服务系统的一种数据处理方法的流程图，该数据处理方法应用于安全服务系统1侧；本公开中的安全服务系统1为预先开发好的能够提供对数据进行安全处理服务的系统，该安全服务系统1至少包括：中断异常处理程序和安全服务函数集合。

[0032] 中断异常处理程序为基于RISC-V架构的处理器进入处理异常状态下所运行的程序，一般通过ecall指令来运行中断异常处理程序，在运行中断异常处理程序时需要中断异常处理程序所实现的操作(功能)，可以根据实际需要来对中断异常处理程序进行预先设计。

[0033] 安全服务函数集合中包括有多个安全服务函数，安全服务函数在被调用运行时，可以所需处理的参数进行相应处理得到处理结果。安全服务函数的数量、以及具体算法可根据实际需要进行预先设计。

[0034] 为保证数据处理过程的安全性，要求用户或外部程序无法对安全服务函数本身以及安全服务函数运行过程中的数据进行更改。为此，在RISC-V架构下，基于PMP安全扩展，可

将安全服务系统所在物理内存区域设置为安全内存区域。

[0035] 如图3所示,该应用于安全服务系统的数据处理方法包括:

[0036] 步骤S101、响应于第一指令,运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数,待处理参数为RTOS所运行任务中的参数。

[0037] 其中,第一指令可以为ecall指令,后文称为第一ecall指令。

[0038] 在位于公共内存区域内的RTOS2所运行任务需要调用安全服务函数进行数据处理时,RTOS2会先将相应参数(即所运行任务中需要通过安全服务函数进行处理的数据)写入至到数据寄存器3中,然后,RTOS2运行第一ecall指令,以表示RTOS2当前所运行任务出现异常,基于RISC-V架构的处理器进入处理异常状态。

[0039] 其中,数据寄存器3为可以存储数据的寄存器。在一些实施例中,数据寄存器3可为整数寄存器或浮点寄存器,整数寄存器为可以存储整数类型数据的寄存器,浮点寄存器为可以存储器浮点类型数据的寄存器。

[0040] 在RTOS2运行第一ecall指令后,响应于第一ecall指令,位于安全内存区域内中断异常处理程序会根据数据寄存器内所存储的待处理参数,来确定待调用的目标安全服务函数。一般地,不同的任务具有不同的任务参数,针对不同的任务参数可选择相应的安全服务函数进行相应处理。换句话说,基于任务参数可以确定所执行的任务类型,针对各类型任务,可以采用相匹配的安全服务函数作相应处理。即,基于任务参数即可确定出对应的目标安全服务函数。

[0041] 步骤S102、运行目标安全服务函数对待处理参数进行处理。

[0042] 目标安全服务函数对待处理参数进行处理,得到运行处理结果,其中运行处理结果可能是单个数值,也可能是由多个数值所构成的数组;具体由目标安全服务函数以及待处理参数来决定。

[0043] 步骤S103、根据目标安全服务函数的运行处理结果将结果信息写入至数据寄存器,以供RTOS从数据寄存器中读取结果信息。

[0044] 在步骤S103中,若处理结果为单个数值时,可将单个数值作为结果信息写入至数据寄存器;若处理结果为由多个数值所构成的数组时,将数组存储至预先设置的共享数据区域,并将数组在共享数据区域内的首地址作为结果信息写入至数据寄存器。

[0045] 在后续过程中,RTOS2从数据寄存器中读取结果信息,从而能够得到RTOS2所运行任务中待处理参数的被安全服务函数进行处理后的处理结果。

[0046] 在本公开实施例中,通过将RTOS2所运行任务中的待处理参数基于RISC-V架构的处理异常机制传递至位于安全内存区域的安全服务系统,位于安全内存区域内的安全服务系统1能够对该待处理参数进行安全处理,有效保证待处理参数的处理过程中不会受到恶意程序的攻击,从而提升了数据处理过程的安全性。

[0047] 图4为本公开实施例中应用于安全服务系统的另一种数据处理方法的流程图,如图4所示,该数据处理方法不但包括前面实施例中的步骤S101～S103,且在步骤S101之前还包括步骤S100a～步骤S100d,且在步骤S103之后还包括:步骤S104;下面仅对步骤S100a～步骤S100d以及步骤S104作详细描述。

[0048] 步骤S100a、初始化中断异常处理程序的入口以及安全服务函数集合内各安全服务函数的入口。

[0049] 步骤S100b、将安全服务系统所在物理内存区域设置为安全内存区域以及将RTOS所在物理内存区域设置为公共内存区域，安全内存区域的读写权限配置为仅读，公共内存区域的读写权限配置为可读且可写。

[0050] 步骤S100c、响应于第二指令，运行中断异常处理程序将mstatus寄存器内MPP域设置用户模式所代表的值，以及将mepc寄存器内的地址修改为RTOS的首地址。

[0051] 步骤S100d、运行第三指令以启动RTOS。

[0052] 其中，第二指令可以为ecall指令，后文称为第二ecall指令；第三指令可以为mret指令，后文称为第二mret指令。

[0053] 再次参见图1所示，在软件开发结束后，使用RISC-V工具链(Toolchain)分别将安全服务系统1所对应的程序以及RTOS2所对应程序(含运行于RTOS上的用户程序)编译成两个elf文件(bin文件也可以)。然后，将这两个elf文件分别加载至基于RISC-V架构的处理器所对应的物理内存区域Q中(例如图1中所示)；其中，需要保证在基于RISC-V架构的处理器上电时，安全服务系统1所对应的elf文件先于RTOS2所对应的elf文件被运行。再接着，基于RISC-V架构的处理器上电，安全服务系统所对应的程序开始运行(从安全服务系统的首地址开始)。

[0054] 此后，执行上述步骤S100a。安全服务系统内设置有初始化模块，初始化模块对中断异常处理程序的入口以及安全服务函数集合内各安全服务函数的入口(即，确定中断异常处理程序的首地址以及各安全服务函数的首地址)初始化，以方便于后续的调用。

[0055] 接着，执行上述步骤S100b；通过PMP表项设置将安全服务系统1所在物理内存区域设置为安全内存区域q1以及将RTOS2所在物理内存区域设置为公共内存区域q2，安全内存区域q1的读写权限配置为仅读，公共内存区域q2的读写权限配置为可读且可写。具体地，可通过PMP表项将安全服务系统所在物理内存区域的特权模式设置为用户模式(User Mod，也称为U模式)，而将RTOS2所在物理内存区域设置为机器模式(Machine Mode，也称为M模式)，以实现对物理内存区域Q的划分。

[0056] 需要说明的是，在本公开实施例中，物理内存区域Q不仅可以划分两块区域，还可以根据需要划分为多块区域，各区域的特权模式可以根据实际需要进行相应设定。

[0057] 再接着，执行上述步骤S100c。安全服务系统自动运行第二ecall指令，根据前面初始化模块所确定出的中断异常处理程序的首地址，以运行中断异常处理程序，中断异常处理程序将mstatus寄存器内MPP域设置用户模式所代表的值，以及将mepc寄存器内的地址修改为RTOS的首地址。

[0058] mstatus寄存器(Machine status register)用于跟踪并控制处理器的当前运行状态；mepc寄存器(Machine exception program Counter)用于记录遇到中断/异常时的地址，以供后续处理中断/异常结束时进行返回。

[0059] 通过上述设置，为后续PC跳转和模式切换提供准备。

[0060] 再接着，开始执行上述步骤S100c；运行第二mret指令实现PC的调整以及模式切换。此时，基于RISC-V架构的处理器会跳转至运行mepc寄存器内的地址，即RTOS的首地址，以启动RTOS。相应地，基于RISC-V架构的处理器处理异常结束。

[0061] 步骤S104、运行第四指令以控制RTOS读取数据寄存器内的结果信息。

[0062] 其中，第四指令可以为mret指令，后文称为第一mret指令

[0063] 相应地，在步骤S103中安全服务系统1将结果信息写入至数据寄存器后，安全服务系统1运行第一mret指令，基于RISC-V架构的处理器会跳转至运行mepc寄存器内所记载的地址，即基于RISC-V架构的处理器遇到中断/异常时的PC值，也就是RTOS2运行第一ecall指令时的PC值，以返回到任务中。其中，PC值是指程序计数器(Program Counter，简称PC)进行计数的值；此后，RTOS会读取数据寄存器内的结果信息，以得到安全服务函数对待处理参数的运行处理结果，然后RTOS继续运行任务。

[0064] 基于同一发明构思，本公开实施例还提供了一种应用于RTOS的数据处理方法。图5为本公开实施例中应用于实时操作系统的一种数据处理方法的流程图，如图5所示，该数据处理方法包括：

[0065] 步骤S201、将所运行任务中的参数作为待处理参数存储至数据寄存器内。

[0066] 步骤S202、运行第一指令，使得安全服务系统运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数，以及运行目标安全服务函数对待处理参数进行处理，并根据目标安全服务函数的运行处理结果将结果信息写入至数据寄存器。

[0067] 步骤S203、从数据寄存器中读取结果信息，并根据结果信息得到运行处理结果。

[0068] 其中，在步骤S203中，当结果信息为单个数值时，将单个数值作为运行处理结果；当结果信息为共享数据区域4内的地址时，根据地址从共享数据区域4内获取由多个数值所构成的数组，并将数组作为运行处理结果。

[0069] 对于上述步骤S201～步骤S203的具体描述，可参见前面关于图1所示实施例中步骤S101～步骤S103的相关内容，此处不再赘述。

[0070] 图6为本公开实施例中应用于实时操作系统的另一种数据处理方法的流程图，如图6所示，该数据处理方法不但包括前面实施例中的步骤S201～S203，且在步骤S201之前还包括步骤S200。下面仅对步骤S200进行详细描述。

[0071] 步骤S200、保存任务内待处理参数的上下文内容。

[0072] 在本公开实施例中，通过保存待处理参数的上下文内容，以供后续安全服务系统1完成对待处理参数的处理后以返回任务时，RTOS2能够根据待处理参数的上下文内容以及对应的处理结果，可以还原得到任务的完整信息。

[0073] 图7为本公开实施例中安全服务系统和实时操作系统实现数据处理方法的一种信令图，如图7所示，该实现数据处理方法包括：

[0074] 步骤S301、安全服务系统初始化中断异常处理程序的入口以及安全服务函数集合内各安全服务函数的入口。

[0075] 步骤S302、安全服务系统通过PMP表项设置将安全服务系统所在物理内存区域设置为安全内存区域以及将RTOS所在物理内存区域设置为公共内存区域，安全内存区域的读写权限配置为仅读，公共内存区域的读写权限配置为可读且可写。

[0076] 步骤S303、安全服务系统响应于第二ecall指令，运行中断异常处理程序将mstatus寄存器内MPP域设置用户模式所代表的值，以及将mepc寄存器内的地址修改为RTOS的首地址。

[0077] 步骤S304、安全服务系统运行第二mret指令以启动RTOS。

[0078] 步骤S305、RTOS保存任务内待处理参数的上下文内容。

[0079] 步骤S306、RTOS将所运行任务中的参数作为待处理参数存储至数据寄存器内。

[0080] 步骤S307、RTOS运行第一ecall指令，以进入中断异常处理程序。

[0081] 步骤S308、安全服务系统响应于第一ecall指令，运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数，待处理参数为RTOS所运行任务中的参数。

[0082] 步骤S309、安全服务系统运行目标安全服务函数对待处理参数进行处理。

[0083] 步骤S310、安全服务系统根据目标安全服务函数的运行处理结果将结果信息写入至数据寄存器。

[0084] 步骤S311、安全服务系统运行第一mret指令以控制RTOS读取数据寄存器内的结果信息。

[0085] 步骤S312、RTOS从数据寄存器中读取结果信息，并根据结果信息得到运行处理结果。

[0086] 图8A为本公开实施例中安全服务系统的一种结构框图，如图8A所示，在安全服务系统1包括：第一运行模块101、第二运行模块102和第一写入模块03。安全服务系统1还包括有中断异常处理程序104和安全服务函数集合105。

[0087] 其中，第一运行模块101配置为响应于第一指令，运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数，待处理参数为RTOS所运行任务中的参数。

[0088] 第二运行模块102配置为运行目标安全服务函数对待处理参数进行处理；

[0089] 第一写入模块103配置为根据目标安全服务函数的运行处理结果将结果信息写入至数据寄存器，以供RTOS从数据寄存器中读取结果信息。

[0090] 图8B为本公开实施例中实时操作系统的一种结构框图，如图8B所示，该RTOS2包括：第二写入模块201、指令运行模块202和读取模块203。

[0091] 其中，第二写入模块201配置为将所运行任务中的参数作为待处理参数写入至数据寄存器内。

[0092] 指令运行模块202配置为运行第一指令，使得安全服务系统运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数，以及运行目标安全服务函数对待处理参数进行处理，并根据目标安全服务函数的运行处理结果将结果信息写入至数据寄存器。

[0093] 读取模块203配置为从数据寄存器中读取结果信息，并根据结果信息得到运行处理结果。

[0094] 对于上述各功能模块的具体描述，可参见前面实施例中关于步骤S101～步骤S103以及步骤S201～步骤S203的相关内容，此处不再赘述。

[0095] 图9为本公开实施例中数据处理系统的一种结构框图，如图9所示，该数据处理系统基于RISC-V框架，具体包括：安全服务系统和RTOS，对于安全服务系统和RTOS的相关描述可参见前面实施例中的内容，此处不再赘述。

[0096] 图10为本公开实施例中提供的电子设备的一种结构框图，如图10所示，该电子设备包括：处理器301、存储器302、一个或多个I/O接口303。存储器302上存储有一个或多个程序，当该一个或多个程序被该一个或多个处理器301执行，使得该一个或多个处理器实现如

上述实施例中任一应用于安全服务系统的数据处理方法中的步骤和/或任一应用于RTOS的数据处理方法中的步骤;一个或多个I/O接口303连接在处理器与存储器之间,配置为实现处理器与存储器的信息交互。

[0097] 其中,处理器301为具有数据处理能力的器件,其包括但不限于中央处理器(CPU)等;存储器302为具有数据存储能力的器件,其包括但不限于随机存取存储器(RAM,更具体如SDRAM、DDR等)、只读存储器(ROM)、带电可擦可编程只读存储器(EEPROM)、闪存(FLASH);I/O接口(读写接口)303连接在处理器301与存储器302间,能实现处理器301与存储器302的信息交互,其包括但不限于数据总线(Bus)等。

[0098] 在一些实施例中,处理器301、存储器302和I/O接口303通过总线304相互连接,进而与计算设备的其它组件连接。

[0099] 在一些实施例中,该处理器301包括FPGA。

[0100] 在一些实施例中,处理器可以为RISC-V处理器。

[0101] 根据本公开的实施例,还提供一种计算机可读介质。该计算机可读介质上存储有计算机程序,其中,该程序被处理器执行时实现如上述实施例任一应用于安全服务系统的数据处理方法中的步骤和/或任一应用于RTOS的数据处理方法中的步骤。

[0102] 特别地,根据本公开实施例,上文参考流程图描述的过程可以被实现为计算机软件程序。例如,本公开的实施例包括一种计算机程序产品,其包括承载在机器可读介质上的计算机程序,该计算机程序包含用于执行流程图所示的方法的程序代码。在这样的实施例中,该计算机程序可以通过通信部分从网络上被下载和安装,和/或从可拆卸介质被安装。在该计算机程序被中央处理单元(CPU)执行时,执行本公开的系统中限定的上述功能。

[0103] 需要说明的是,本公开所示的计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质或者是上述两者的任意组合。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子可以包括但不限于:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本公开中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。而在本公开中,计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括但不限于:无线、电线、光缆、RF等等,或者上述的任意合适的组合。

[0104] 附图中的流程图和框图,图示了按照本公开各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分,前述模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所

标注的功能也可以以不同于附图中所标注的顺序发生。例如，两个接连地表示的方框实际上可以基本并行地执行，它们有时也可以按相反的顺序执行，这依所涉及的功能而定。也要注意的是，框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合，可以用执行规定的功能或操作的专用的基于硬件的系统来实现，或者可以用专用硬件与计算机指令的组合来实现。

[0105] 可以理解的是，以上实施方式仅仅是为了说明本发明的原理而采用的示例性实施方式，然而本发明并不局限于此。对于本领域内的普通技术人员而言，在不脱离本发明的精神和实质的情况下，可以做出各种变型和改进，这些变型和改进也视为本发明的保护范围。

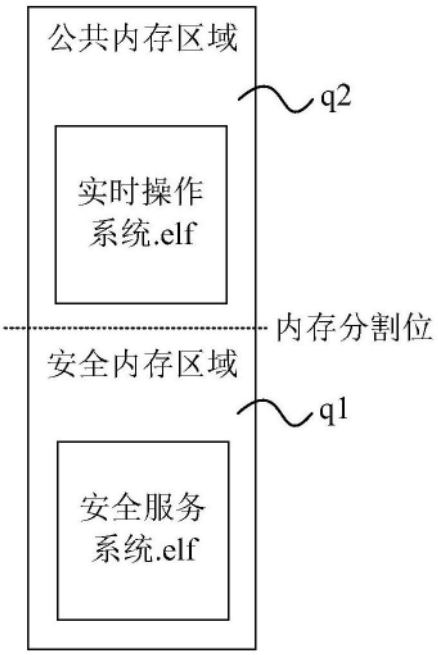

公共内存区域
q2
实时操作
系统.elf
内存分割位
安全内存区域
q1
安全服务
系统.elf

图1

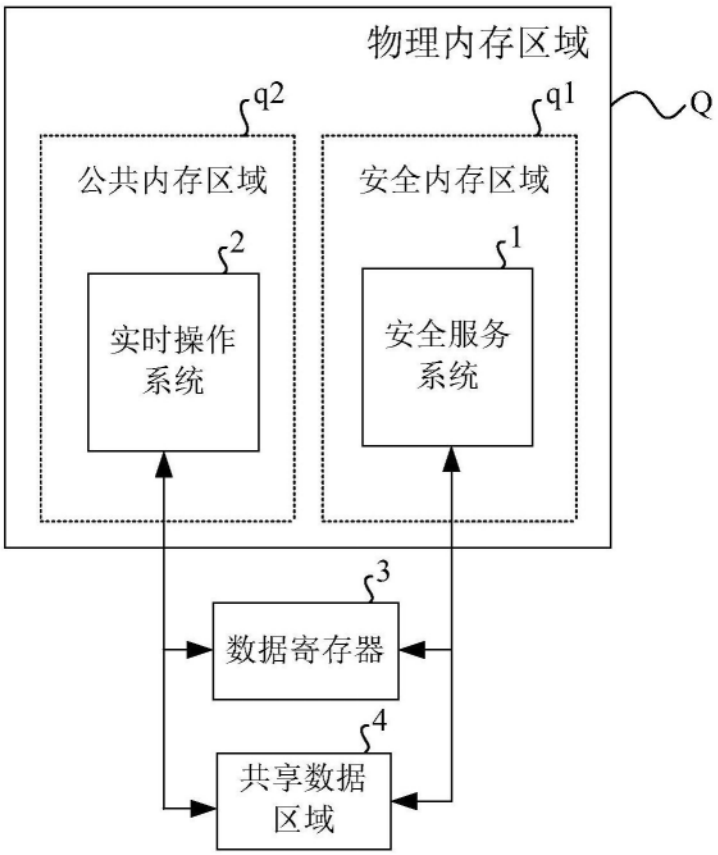

物理内存区域
q2
q1
Q
公共内存区域
安全内存区域
2
1
实时操作
系统
安全服务
系统
3
数据寄存器
4
共享数据
区域

图2

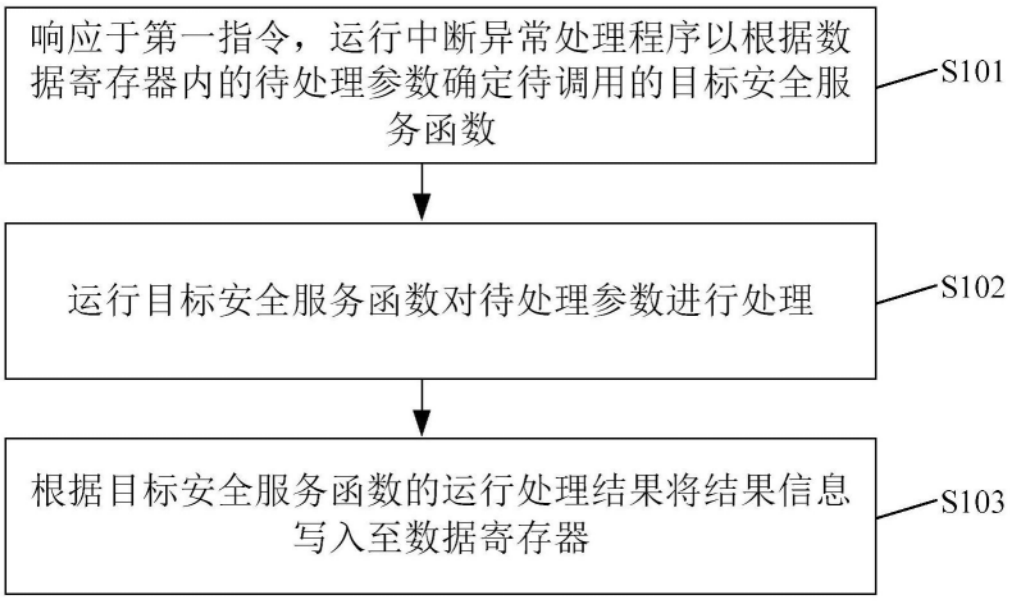

响应于第一指令，运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数
S101
运行目标安全服务函数对待处理参数进行处理
S102
根据目标安全服务函数的运行处理结果将结果信息写入至数据寄存器
S103

图3

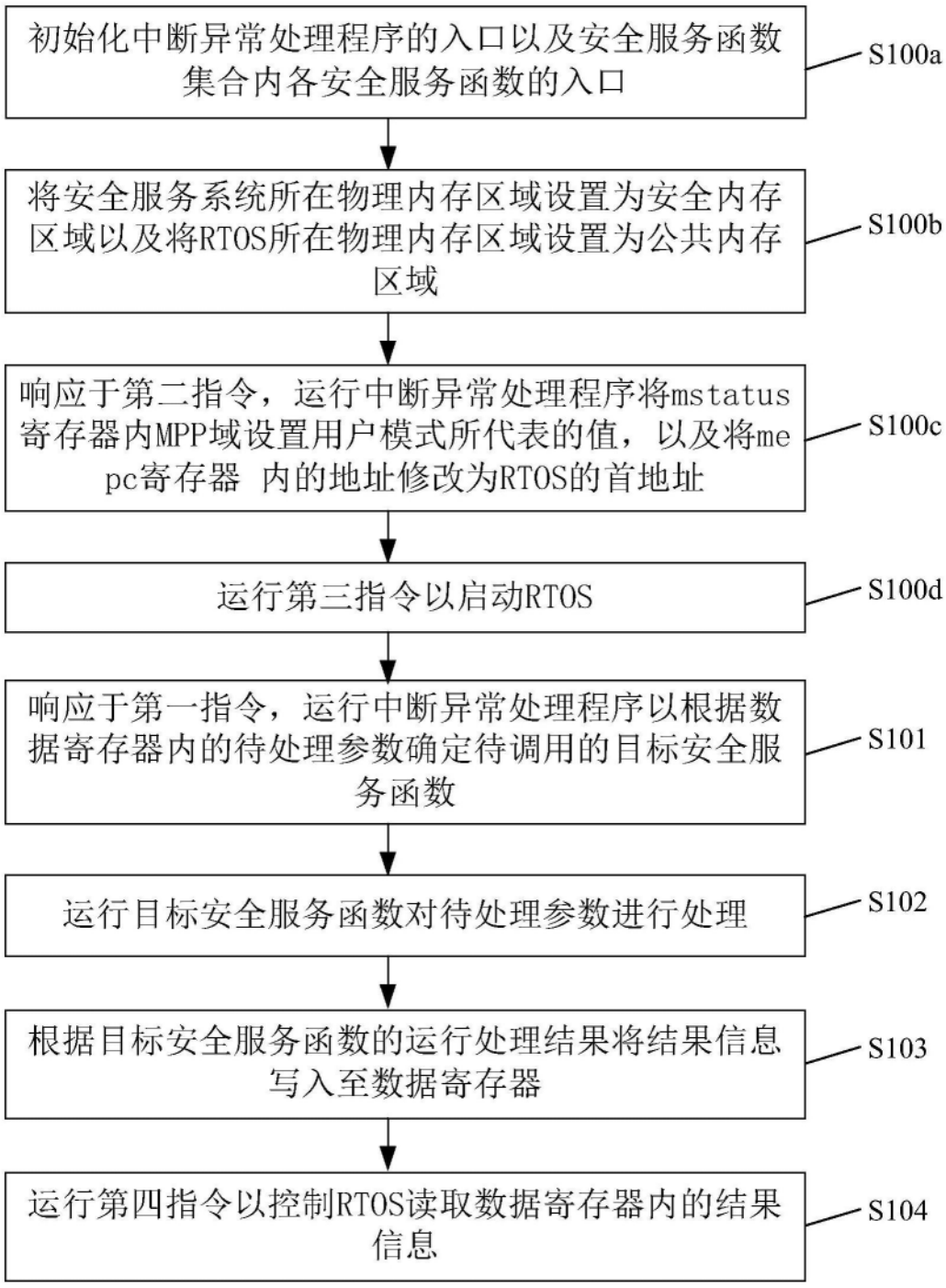

初始化中断异常处理程序的入口以及安全服务函数集合内各安全服务函数的入口
S100a
将安全服务系统所在物理内存区域设置为安全内存区域以及将RTOS所在物理内存区域设置为公共内存区域
S100b
响应于第二指令，运行中断异常处理程序将mstatus寄存器内MPP域设置用户模式所代表的值，以及将mepc寄存器 内的地址修改为RTOS的首地址
S100c
运行第三指令以启动RTOS
S100d
响应于第一指令，运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数
S101
运行目标安全服务函数对待处理参数进行处理
S102
根据目标安全服务函数的运行处理结果将结果信息写入至数据寄存器
S103
运行第四指令以控制RTOS读取数据寄存器内的结果信息
S104

图4

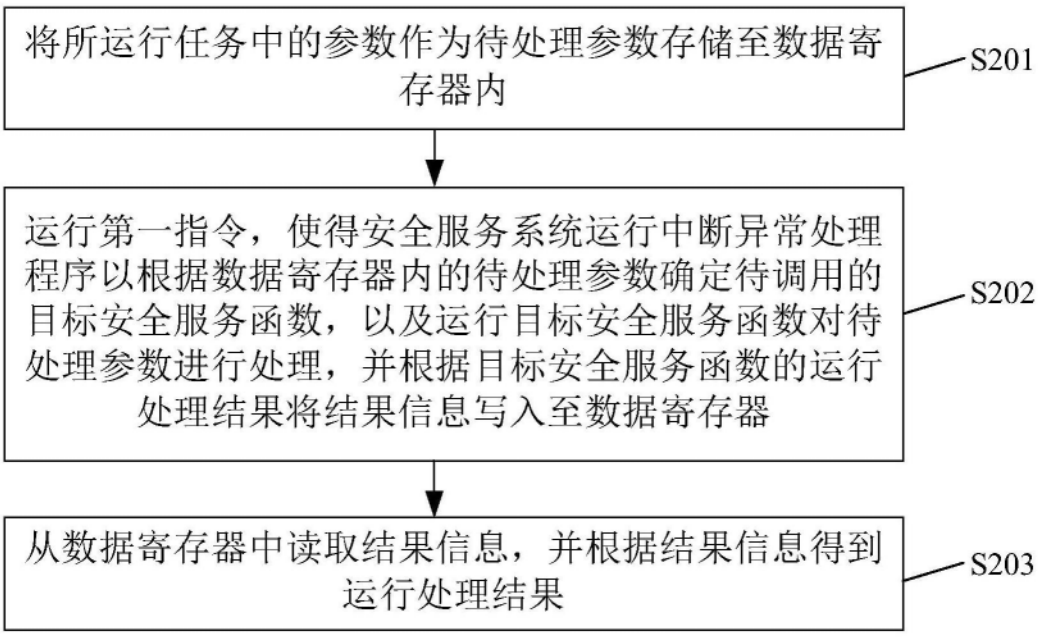

将所运行任务中的参数作为待处理参数存储至数据寄存器内
S201
运行第一指令，使得安全服务系统运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数，以及运行目标安全服务函数对待处理参数进行处理，并根据目标安全服务函数的运行处理结果将结果信息写入至数据寄存器
S202
从数据寄存器中读取结果信息，并根据结果信息得到运行处理结果
S203

图5

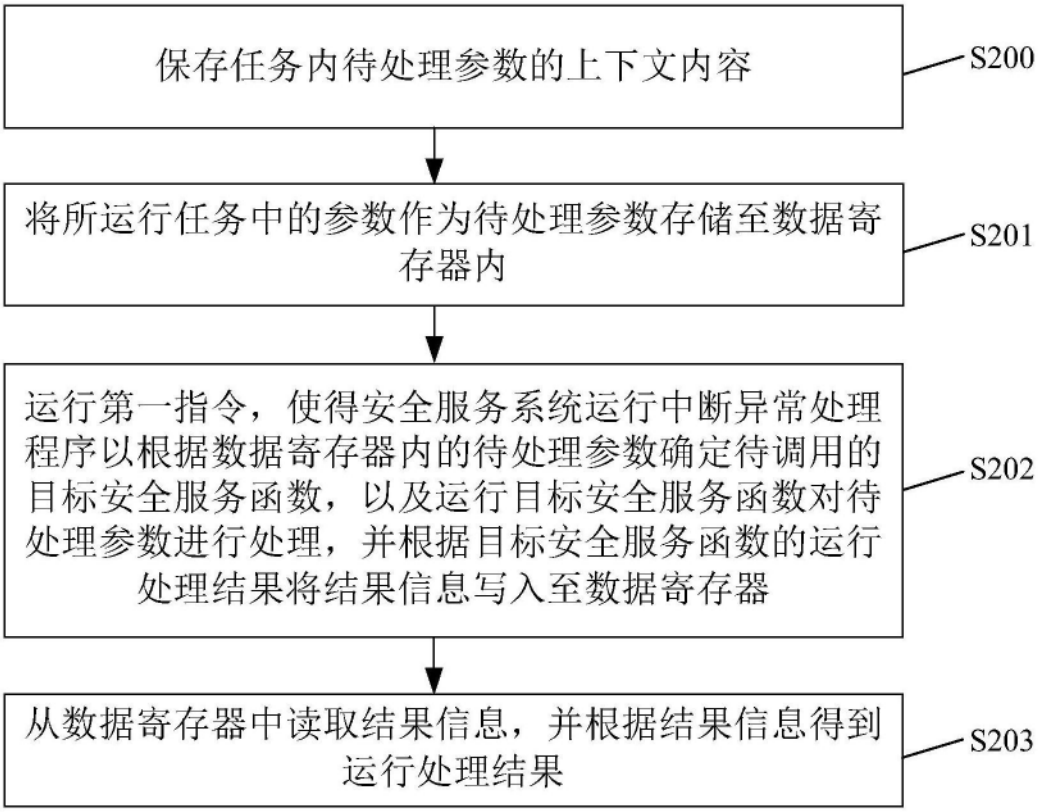

保存任务内待处理参数的上下文内容
S200
将所运行任务中的参数作为待处理参数存储至数据寄存器内
S201
运行第一指令，使得安全服务系统运行中断异常处理程序以根据数据寄存器内的待处理参数确定待调用的目标安全服务函数，以及运行目标安全服务函数对待处理参数进行处理，并根据目标安全服务函数的运行处理结果将结果信息写入至数据寄存器
S202
从数据寄存器中读取结果信息，并根据结果信息得到运行处理结果
S203

图6

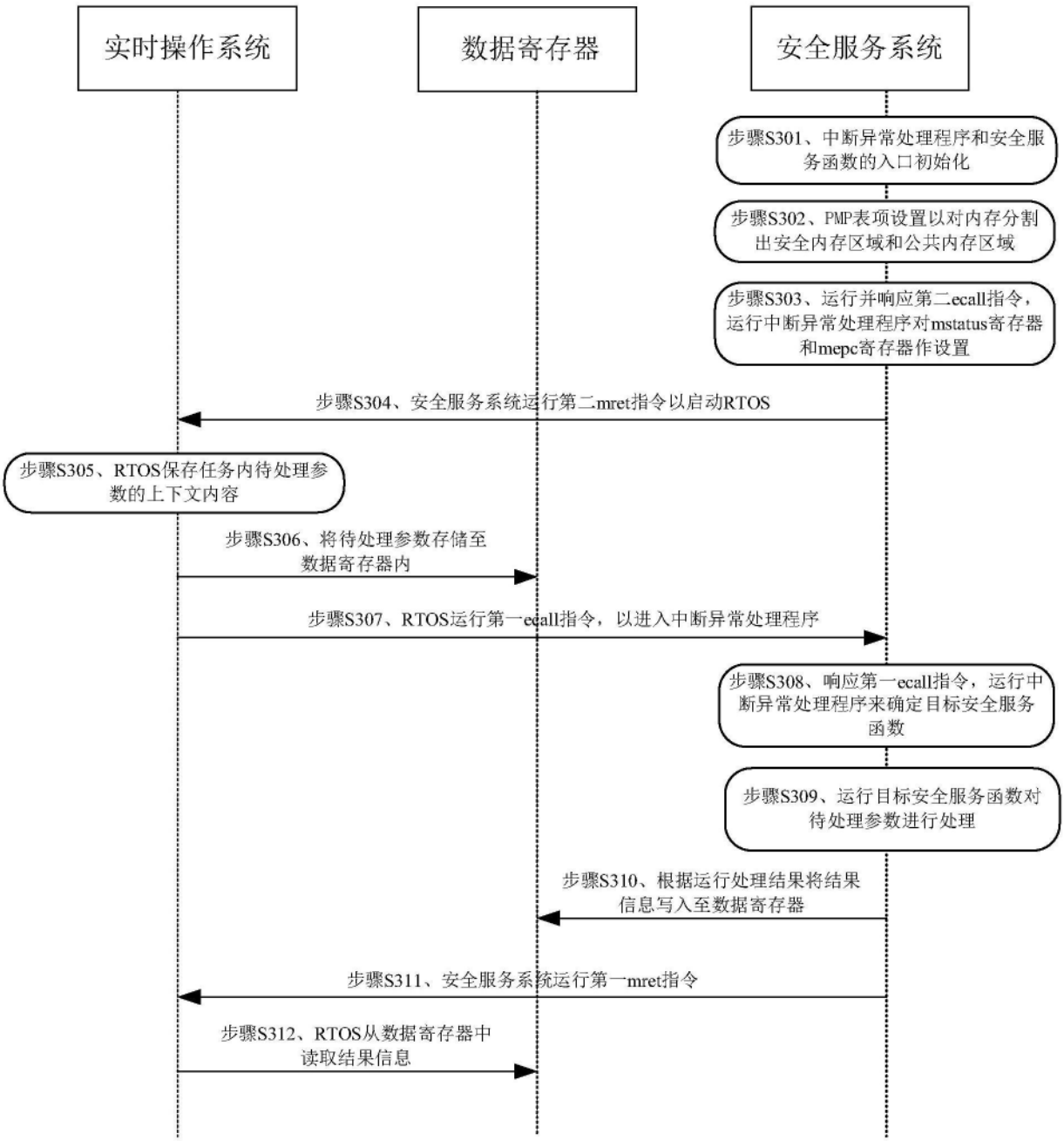

实时操作系统
数据寄存器
安全服务系统
步骤S301、中断异常处理程序和安全服务函数的入口初始化
步骤S302、PMP表项设置以对内存分割出安全内存区域和公共内存区域
步骤S303、运行并响应第二ecall指令，运行中断异常处理程序对mstatus寄存器和mepc寄存器作设置
步骤S304、安全服务系统运行第二mret指令以启动RTOS
步骤S305、RTOS保存任务内待处理参数的上下文内容
步骤S306、将待处理参数存储至数据寄存器内
步骤S307、RTOS运行第一ecall指令，以进入中断异常处理程序
步骤S308、响应第一ecall指令，运行中断异常处理程序来确定目标安全服务函数
步骤S309、运行目标安全服务函数对待处理参数进行处理
步骤S310、根据运行处理结果将结果信息写入至数据寄存器
步骤S311、安全服务系统运行第一mret指令
步骤S312、RTOS从数据寄存器中读取结果信息

图7

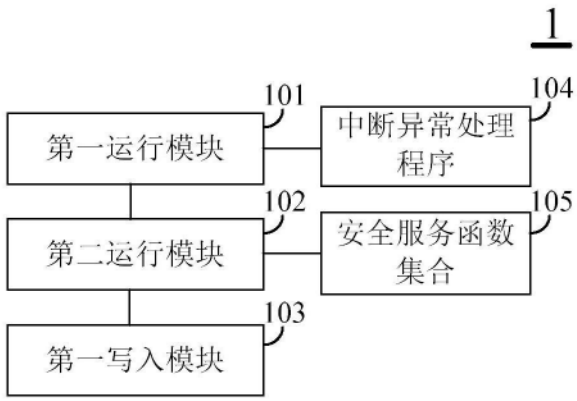

1
101
第一运行模块
104
中断异常处理程序
102
第二运行模块
105
安全服务函数集合
103
第一写入模块

图8A

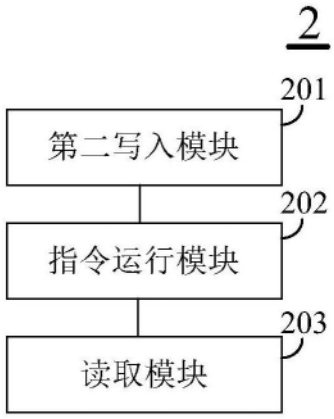

2
201
第二写入模块
202
指令运行模块
203
读取模块

图8B

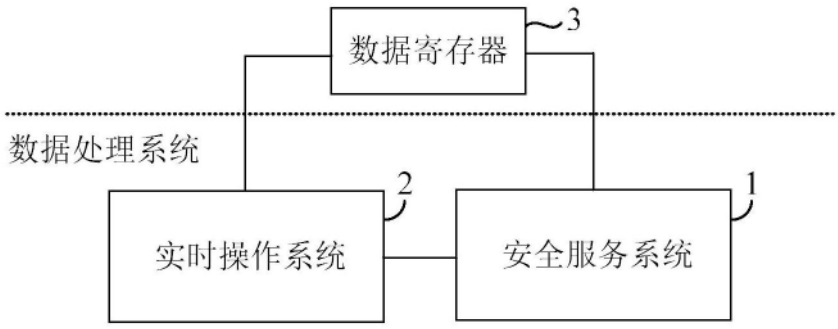

3
数据寄存器
数据处理系统
2
实时操作系统
1
安全服务系统

图9

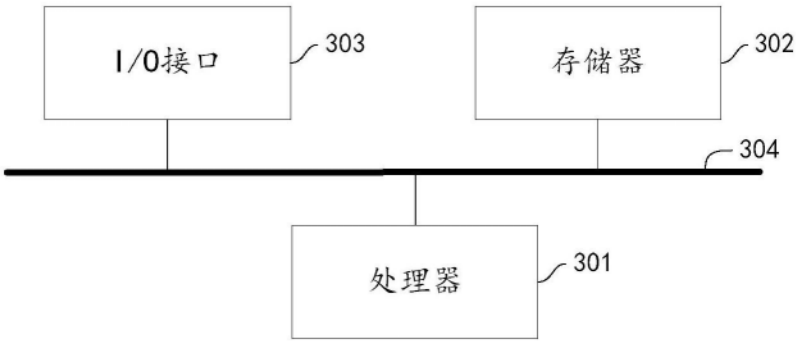


图10