



(12) 发明专利申请

(10) 申请公布号 CN 113986476 A

(43) 申请公布日 2022. 01. 28

(21) 申请号 202111592516.6

(22) 申请日 2021.12.24

(71) 申请人 北京鲸鲮信息系统技术有限公司
地址 100080 北京市海淀区西大街39号2层
2005

(72) 发明人 邓宝安

(74) 专利代理机构 北京市铸成律师事务所
11313

代理人 包莉莉 曹远

(51) Int. Cl.

G06F 9/455 (2006.01)

G06F 9/54 (2006.01)

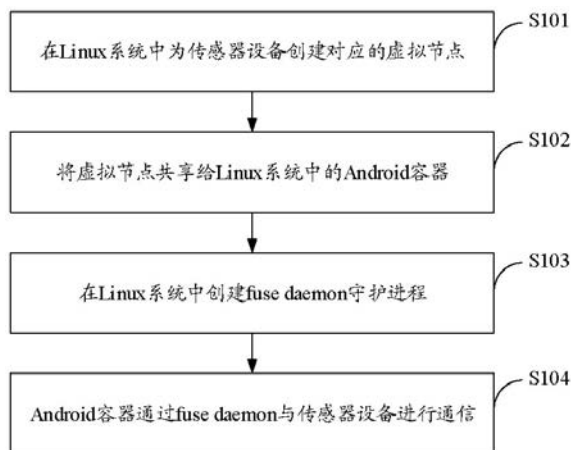
权利要求书2页 说明书7页 附图4页

(54) 发明名称

传感器设备虚拟化的方法、装置、电子设备和存储介质

(57) 摘要

本公开提供了一种传感器设备虚拟化的方法、装置、电子设备和存储介质,涉及计算机技术领域。该方法包括:在Linux系统中为传感器设备创建对应的虚拟节点,将所述虚拟节点共享给所述Linux系统中的Android容器,在所述Linux系统中创建fuse daemon守护进程,所述Android容器通过所述fuse daemon与所述传感器设备进行通信。本公开技术方案通过对传感器设备进行虚拟化,实现主机Linux系统对传感器设备的统一管理,可以有效地解决Linux系统中运行Android容器时传感器设备的兼容问题。



1. 一种传感器设备虚拟化的方法,包括:
在Linux系统中为传感器设备创建对应的虚拟节点;
将所述虚拟节点共享给所述Linux系统中的Android容器;
在所述Linux系统中创建fuse daemon守护进程;
所述Android容器通过所述fuse daemon与所述传感器设备进行通信。
2. 根据权利要求1所述的方法,其中,所述在Linux系统中为传感器设备创建对应的虚拟节点,包括:
在Linux系统中使用fuse用户空间文件系统,为传感器设备创建对应的虚拟节点。
3. 根据权利要求2所述的方法,其中,所述在Linux系统中使用fuse用户空间文件系统,为传感器设备创建对应的虚拟节点,包括:
为传感器设备创建fuse文件夹,使用mount挂载函数将所述fuse文件夹挂载到目录/dev/fuse下,生成所述传感器设备对应的虚拟节点。
4. 根据权利要求1所述的方法,其中,所述将所述虚拟节点共享给所述Linux系统中的Android容器,包括:
通过mount挂载函数将所述虚拟节点挂载到所述Linux系统中的Android容器中,对所述Android容器可见。
5. 根据权利要求1所述的方法,还包括:
将HAL硬件抽象层中注册的所述传感器设备修改为所述虚拟节点。
6. 根据权利要求5所述的方法,所述Android容器通过所述fuse daemon与所述传感器设备进行通信,包括:
所述Android容器基于所述HAL硬件抽象层调用所述传感器设备的接口函数时,所述fuse daemon通过fuse_operations用户空间文件系统操作接口与所述Linux系统内的传感器服务器通信,完成对所述传感器设备的输入或输出操作。
7. 一种传感器设备虚拟化的装置,包括:
虚拟模块,用于在Linux系统中为传感器设备创建对应的虚拟节点;
共享模块,用于将所述虚拟节点共享给所述Linux系统中的Android容器;
创建模块,用于在所述Linux系统中创建fuse daemon守护进程;
通信模块,用于所述Android容器通过所述fuse daemon与所述传感器设备进行通信。
8. 根据权利要求7所述的装置,其中,所述虚拟模块包括:
虚拟单元,用于在Linux系统中使用fuse用户空间文件系统,为传感器设备创建对应的虚拟节点。
9. 根据权利要求8所述的装置,其中,所述虚拟单元用于:
为传感器设备创建fuse文件夹,使用mount挂载函数将所述fuse文件夹挂载到目录/dev/fuse下,生成所述传感器设备对应的虚拟节点。
10. 根据权利要求7所述的装置,其中,所述共享模块用于:
通过mount挂载函数将所述虚拟节点挂载到所述Linux系统中的Android容器中,对所述Android容器可见。
11. 根据权利要求7所述的装置,还包括:
修改模块,用于将HAL硬件抽象层中注册的所述传感器设备修改为所述虚拟节点。

12. 根据权利要求11所述的装置,所述通信模块用于:

所述Android容器基于所述HAL硬件抽象层调用所述传感器设备的接口函数时,所述fuse daemon通过fuse_operations用户空间文件系统操作接口与所述Linux系统内的传感器服务器通信,完成对所述传感器设备的输入或输出操作。

13. 一种电子设备,包括:

至少一个处理器;以及

与所述至少一个处理器通信连接的存储器;其中,

所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行权利要求1-6中任一项所述的方法。

14. 一种存储有计算机指令的非瞬时计算机可读存储介质,其中,所述计算机指令用于使所述计算机执行根据权利要求1-6中任一项所述的方法。

传感器设备虚拟化的方法、装置、电子设备和存储介质

技术领域

[0001] 本公开涉及计算机技术领域,尤其涉及Linux和Android兼容技术领域。

背景技术

[0002] 目前,Linux系统普遍通过容器来实现Android系统的兼容,如通过Anbox(Android in a box,容器中的安卓),使用Linux命名空间在容器中运行完整的Android系统。当Linux系统内的Android容器访问传感器sensor设备时,可以通过目录/dev/下对应该传感器设备的节点共享的方式进行访问。

[0003] 但是,共享的方式下传感器设备只能由Linux系统或Android容器一方使用,Android容器对于传感器设备的访问不受Linux系统管理,如果Linux系统和Android容器同时访问传感器设备,则会产生冲突。

发明内容

[0004] 本公开提供了一种传感器设备虚拟化的方法、装置、电子设备、存储介质和计算机程序产品。

[0005] 根据本公开的一方面,提供了一种传感器设备虚拟化的方法,包括:

在Linux系统中为传感器设备创建对应的虚拟节点;

将所述虚拟节点共享给所述Linux系统中的Android容器;

在所述Linux系统中创建fuse daemon守护进程;

所述Android容器通过所述fuse daemon与所述传感器设备进行通信。

[0006] 根据本公开的另一方面,提供了一种传感器设备虚拟化的装置,包括:

虚拟模块,用于在Linux系统中为传感器设备创建对应的虚拟节点;

共享模块,用于将所述虚拟节点共享给所述Linux系统中的Android容器;

创建模块,用于在所述Linux系统中创建fuse daemon守护进程;

通信模块,用于所述Android容器通过所述fuse daemon与所述传感器设备进行通信。

[0007] 根据本公开的另一方面,提供了一种电子设备,包括:

至少一个处理器;以及

与所述至少一个处理器通信连接的存储器;其中,

所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行本公开任一实施例中的方法。

[0008] 根据本公开的另一方面,提供了一种存储有计算机指令的非瞬时计算机可读存储介质,其中,所述计算机指令用于使所述计算机执行本公开任一实施例中的方法。

[0009] 根据本公开的另一方面,提供了一种计算机程序产品,包括计算机程序,所述计算机程序在被处理器执行时实现本公开任一实施例中的方法。

[0010] 本公开实施例的技术方案,通过对传感器设备进行虚拟化,实现主机Linux系统对

传感器设备的统一管理,可以有效地解决Linux系统中运行Android容器时传感器设备的兼容问题。

[0011] 应当理解,本部分所描述的内容并非旨在标识本公开的实施例的关键或重要特征,也不用于限制本公开的范围。本公开的其它特征将通过以下的说明书而变得容易理解。

附图说明

[0012] 附图用于更好地理解本方案,不构成对本公开的限定。其中:
图1是根据本公开一实施例中传感器设备虚拟化的方法示意图;
图2是根据本公开一实施例中传感器设备虚拟化的方法示意图;
图3是根据本公开一实施例中Linux系统内部框架示意图;
图4是根据本公开一实施例中传感器设备虚拟化的装置框图;
图5是根据本公开一实施例中传感器设备虚拟化的装置框图;
图6是用来实现本公开实施例的传感器设备虚拟化的方法的电子设备的框图。

具体实施方式

[0013] 以下结合附图对本公开的示范性实施例做出说明,其中包括本公开实施例的各种细节以助于理解,应当将它们认为仅仅是示范性的。因此,本领域普通技术人员应当认识到,可以对这里描述的实施例做出各种改变和修改,而不会背离本公开的范围和精神。同样,为了清楚和简明,以下的描述中省略了对公知功能和结构的描述。

[0014] 本公开实施例的技术方案应用于Linux系统和Android系统同时运行的场景,其中,Android系统以容器的方式在主机Linux系统中运行,该主机Linux系统可以连接一个或多个传感器设备。本公开实施例涉及的传感器设备可以有多种类型,包括但不限于:温度传感器、湿度传感器、压力传感器、位置传感器、速度传感器、加速度传感器和热敏传感器等等。Linux系统或Android系统与传感器设备的交互可以是双向操作,包括输入操作和输出操作。其中,输入操作是指Linux系统或Android系统可以从传感器设备获取数据,如获取温度、湿度和压力等数据。输出操作是指Linux系统或Android系统可以控制传感器设备的参数,如调节传感器的灵敏度、精度、频率范围或误差范围等等。

[0015] 本公开实施例通过对传感器设备进行虚拟化,实现主机Linux系统对传感器设备的统一管理,可以有效地解决Linux系统中运行Android容器时传感器设备的兼容问题。

[0016] 图1为本公开一实施例中传感器设备虚拟化的方法示意图。如图1所示,该方法包括:

- S101:在Linux系统中为传感器设备创建对应的虚拟节点;
- S102:将虚拟节点共享给Linux系统中的Android容器;
- S103:在Linux系统中创建fuse daemon守护进程;
- S104:Android容器通过fuse daemon与传感器设备进行通信。

[0017] 在一种实施方式中,在Linux系统中为传感器设备创建对应的虚拟节点,包括:
在Linux系统中使用fuse用户空间文件系统,为传感器设备创建对应的虚拟节点。

[0018] 本公开实施例中,fuse(Filesystem in Userspace,用户空间文件系统)是Linux中用于挂载某些网络空间到本地文件系统的模块,是完全在用户态实现的文件系统。Linux

从2.6.14版本开始通过fuse支持在用户空间实现文件系统。在用户空间实现文件系统能够大幅提高生产率,简化了为操作系统提供新的文件系统的工作量,特别适用于各种虚拟文件系统 and 网络文件系统。Fuse驱动程序可以视为将请求重定向到后端用户守护程序的代理,通过fuse可以很容易的虚拟出不同的设备文件。

[0019] 在一种实施方式中,在Linux系统中使用fuse用户空间文件系统,为传感器设备创建对应的虚拟节点,包括:

为传感器设备创建fuse文件夹,使用mount挂载函数将fuse文件夹挂载到目录/dev/fuse下,生成传感器设备对应的虚拟节点。

[0020] 本公开实施例兼顾传感器设备双向数据传输以及Linux系统和Android系统同时访问传感器设备互斥的特点,使用fuse来实现传感器设备的虚拟化,不仅能够实现传输虚拟事件给Android容器,而且能够实现Android容器对传感器设备的各种操作。

[0021] 在一种实施方式中,将虚拟节点共享给Linux系统中的Android容器,包括:

通过mount挂载函数将虚拟节点挂载到Linux系统中的Android容器中,对Android容器可见。

[0022] 在一种实施方式中,上述方法还包括:

将HAL硬件抽象层中注册的传感器设备修改为虚拟节点。

[0023] 通过修改HAL可以使Android容器访问的设备节点改为虚拟节点,起到了隔离传感器设备的效果,以访问虚拟节点的方式实现了Android容器对传感器设备的访问。

[0024] 在一种实施方式中,Android容器通过fuse daemon与传感器设备进行通信,包括:

Android容器基于HAL调用传感器设备的接口函数时,fuse daemon通过fuse_operations用户空间文件系统操作接口与Linux系统内的传感器服务器通信,完成对传感器设备的输入或输出操作。

[0025] 本公开实施例中,fuse_operations为一组接口,包括多个操作传感器设备的接口,如open(), read(), write(), ioctl()和poll()等等,这些接口通过dbus总线直接与Linux系统内的传感器服务器sensor server通讯,用于控制和获取传感器设备的数据。

[0026] 通过fuse daemon实现了fuse_operations接口的调用,保证了Android容器与传感器设备的通信,实现了Android容器对传感器设备的各种操作。

[0027] 本公开实施例提供的上述方法,通过对传感器设备进行虚拟化,实现主机Linux系统对传感器设备的统一管理,Linux系统与传感器设备直接通信,Android容器以访问虚拟节点的方式与传感器设备进行通信,避免了Linux系统和Android容器同时访问传感器设备时产生冲突,有效地解决了Linux系统中运行Android容器时传感器设备的兼容问题。

[0028] 图2为本公开一实施例中传感器设备虚拟化的方法示意图。如图2所示,该方法包括:

S201:在Linux系统中使用fuse用户空间文件系统,为传感器设备创建对应的虚拟节点;

本公开实施例中,上述步骤S201可以包括:

为传感器设备创建fuse文件夹,使用mount挂载函数将fuse文件夹挂载到目录/dev/fuse下,生成传感器设备对应的虚拟节点。

[0029] 其中,为传感器设备创建fuse文件夹可以参考传感器HAL中支持的传感器类型来

创建,具体不限定。

[0030] S202:通过mount挂载函数将虚拟节点挂载到Linux系统中的Android容器中,对Android容器可见;

S203:将HAL硬件抽象层中注册的传感器设备修改为虚拟节点;

在一种实施方式中,可以修改HAL(Hardware Abstraction Layer)中的sensor_open,将其注册的传感器设备修改为虚拟节点对应的fuse文件,从而保证了HAL调用open(), read(), write(), ioctl()或poll()等函数时,可以由fuse daemon捕获到。

[0031] S204:在Linux系统中创建fuse daemon守护进程;

本公开实施例中,fuse daemon的作用是在Android容器访问虚拟节点时,通过fuse daemon能够访问传感器服务器sensor server,进而实现控制和获取传感器设备的数据。上述将虚拟节点对应的fuse文件夹挂载到目录/dev/fuse下以后,对该虚拟节点对应的fuse文件访问均会由fuse daemon获取并处理。

[0032] S205:Android容器基于HAL调用传感器设备的接口函数时,fuse daemon通过fuse_operations用户空间文件系统操作接口与Linux系统内的传感器服务器通信,完成对传感器设备的输入或输出操作。

[0033] 在一种实施方式中,针对Android容器对虚拟节点的调用,相应地,fuse daemon可以实现fuse_operations中如open(), read(), write(), ioctl()或poll()等各种接口的调用,这些接口通过dbus总线直接与Linux系统内的传感器服务器sensor server通讯,实现控制和获取传感器设备数据。

[0034] 图3为本公开一实施例中Linux系统内部框架示意图。如图3所示,在一种实施方式中,使用Ixc方法在Linux系统中运行Android容器。在Linux系统中使用fuse创建传感器设备对应的虚拟节点,生成的fuse文件夹挂载到目录/dev/fuse下,且以容器共享的方式挂载到Android容器中。将HAL中注册的传感器设备修改为虚拟节点。在Linux系统中创建fuse daemon守护进程。Android容器基于HAL调用传感器设备的接口函数时,fuse daemon通过fuse_operations接口与Linux系统内的传感器服务器sensorfw server通信,完成对传感器设备的输入或输出操作。

[0035] 本公开实施例提供的上述方法,通过对传感器设备进行虚拟化,实现主机Linux系统对传感器设备的统一管理,Linux系统与传感器设备直接通信,Android容器以访问虚拟节点的方式与传感器设备进行通信,避免了Linux系统和Android容器同时访问传感器设备时产生冲突,有效地解决了Linux系统中运行Android容器时传感器设备的兼容问题。

[0036] 图4是根据本公开一实施例中传感器设备虚拟化的装置框图。如图4所示,该装置包括:

虚拟模块401,用于在Linux系统中为传感器设备创建对应的虚拟节点;

共享模块402,用于将虚拟节点共享给Linux系统中的Android容器;

创建模块403,用于在Linux系统中创建fuse daemon守护进程;

通信模块404,用于Android容器通过fuse daemon与传感器设备进行通信。

[0037] 在一种实施方式中,虚拟模块401包括:

虚拟单元,用于在Linux系统中使用fuse用户空间文件系统,为传感器设备创建对应的虚拟节点。

[0038] 在一种实施方式中,虚拟单元用于:

为传感器设备创建fuse文件夹,使用mount挂载函数将fuse文件夹挂载到目录/dev/fuse下,生成传感器设备对应的虚拟节点。

[0039] 在一种实施方式中,共享模块402用于:

通过mount挂载函数将虚拟节点挂载到Linux系统中的Android容器中,对Android容器可见。

[0040] 图5是根据本公开一实施例中的装置框图。如图5所示,该装置包括:虚拟模块501、共享模块502、创建模块503、通信模块504和修改模块505。

[0041] 其中,虚拟模块501、共享模块502、创建模块503和通信模块504分别与上述实施例中的虚拟模块401、共享模块402、创建模块403和通信模块404功能相同,此处不赘述。

[0042] 在一种实施方式中,修改模块505用于将HAL硬件抽象层中注册的传感器设备修改为虚拟节点。

[0043] 在一种实施方式中,通信模块504用于:

Android容器基于HAL调用传感器设备的接口函数时,fuse daemon通过fuse_operations用户空间文件系统操作接口与Linux系统内的传感器服务器通信,完成对传感器设备的输入或输出操作。

[0044] 本公开实施例提供的上述装置,通过对传感器设备进行虚拟化,实现主机Linux系统对传感器设备的统一管理,Linux系统与传感器设备直接通信,Android容器以访问虚拟节点的方式与传感器设备进行通信,避免了Linux系统和Android容器同时访问传感器设备时产生冲突,有效地解决了Linux系统中运行Android容器时传感器设备的兼容问题。

[0045] 根据本公开的实施例,本公开还提供了一种电子设备、一种可读存储介质和一种计算机程序产品。

[0046] 图6示出了可以用来实施本公开的实施例的示例电子设备600的示意性框图。电子设备旨在表示各种形式的数字计算机,诸如,膝上型计算机、台式计算机、工作台、个人数字助理、服务器、刀片式服务器、大型计算机、和其它适合的计算机。电子设备还可以表示各种形式的移动装置,诸如,个人数字处理、蜂窝电话、智能电话、可穿戴设备和其它类似的计算装置。本文所示的部件、它们的连接和关系、以及它们的功能仅仅作为示例,并且不意在限制本文中描述的和/或者要求的本公开的实现。

[0047] 如图6所示,设备600包括计算单元601,其可以根据存储在只读存储器(ROM) 602中的计算机程序或者从存储单元608加载到随机访问存储器(RAM) 603中的计算机程序,来执行各种适当的动作和处理。在RAM 603中,还可存储设备600操作所需的各种程序和数据。计算单元601、ROM 602以及RAM 603通过总线604彼此相连。输入/输出(I/O)接口605也连接至总线604。

[0048] 设备600中的多个部件连接至I/O接口605,包括:输入单元606,例如键盘、鼠标等;输出单元607,例如各种类型的显示器、扬声器等;存储单元608,例如磁盘、光盘等;以及通信单元609,例如网卡、调制解调器、无线通信收发机等。通信单元609允许设备600通过诸如因特网的计算机网络和/或各种电信网络与其他设备交换信息/数据。

[0049] 计算单元601可以是各种具有处理和计算能力的通用和/或专用处理组件。计算单元601的一些示例包括但不限于中央处理单元(CPU)、图形处理单元(GPU)、各种专用的人工

智能(AI)计算芯片、各种运行机器学习模型算法的计算单元、数字信号处理器(DSP)、以及任何适当的处理器、控制器、微控制器等。计算单元601执行上文所描述的各个方法和处理。例如,在一些实施例中,上述方法可被实现为计算机软件程序,其被有形地包含于机器可读介质,例如存储单元608。在一些实施例中,计算机程序的部分或者全部可以经由ROM 602和/或通信单元609而被载入和/或安装到设备600上。当计算机程序加载到RAM 603并由计算单元601执行时,可以执行上文描述的方法的一个或多个步骤。备选地,在其他实施例中,计算单元601可以通过其他任何适当的方式(例如,借助于固件)而被配置为执行上述方法。

[0050] 本文中以上描述的系统和技术各种实施方式可以在数字电子电路系统、集成电路系统、场可编程门阵列(FPGA)、专用集成电路(ASIC)、专用标准产品(ASSP)、芯片上系统的系统(SOC)、负载可编程逻辑设备(CPLD)、计算机硬件、固件、软件、和/或它们的组合中实现。这些各种实施方式可以包括:实施在一个或者多个计算机程序中,该一个或者多个计算机程序可在包括至少一个可编程处理器的可编程系统上执行和/或解释,该可编程处理器可以是专用或者通用可编程处理器,可以从存储系统、至少一个输入装置、和至少一个输出装置接收数据和指令,并且将数据和指令传输至该存储系统、该至少一个输入装置、和该至少一个输出装置。

[0051] 用于实施本公开的方法的程序代码可以采用一个或多个编程语言的任何组合来编写。这些程序代码可以提供给通用计算机、专用计算机或其他可编程数据处理装置的处理器或控制器,使得程序代码当由处理器或控制器执行时使流程图和/或框图中所规定的功能/操作被实施。程序代码可以完全在机器上执行、部分地在机器上执行,作为独立软件包部分地在机器上执行且部分地在远程机器上执行或完全在远程机器或服务器上执行。

[0052] 在本公开的上下文中,机器可读介质可以是有形的介质,其可以包含或存储以供指令执行系统、装置或设备使用或与指令执行系统、装置或设备结合地使用的程序。机器可读介质可以是机器可读信号介质或机器可读储存介质。机器可读介质可以包括但不限于电子的、磁性的、光学的、电磁的、红外的、或半导体系统、装置或设备,或者上述内容的任何合适组合。机器可读存储介质的更具体示例会包括基于一个或多个线的电气连接、便携式计算机盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦除可编程只读存储器(EPROM或快闪存储器)、光纤、便捷式紧凑盘只读存储器(CD-ROM)、光学储存设备、磁储存设备、或上述内容的任何合适组合。

[0053] 为了提供与用户的交互,可以在计算机上实施此处描述的系统和技术,该计算机具有:用于向用户显示信息的显示装置(例如,CRT(阴极射线管)或者LCD(液晶显示器)监视器);以及键盘和指向装置(例如,鼠标或者轨迹球),用户可以通过该键盘和该指向装置来将输入提供给计算机。其它种类的装置还可以用于提供与用户的交互;例如,提供给用户的反馈可以是任何形式的传感反馈(例如,视觉反馈、听觉反馈、或者触觉反馈);并且可以用任何形式(包括声输入、语音输入或者、触觉输入)来接收来自用户的输入。

[0054] 可以将此处描述的系统和技术实施在包括后台部件的计算系统(例如,作为数据服务器)、或者包括中间件部件的计算系统(例如,应用服务器)、或者包括前端部件的计算系统(例如,具有图形用户界面或者网络浏览器的用户计算机,用户可以通过该图形用户界面或者该网络浏览器来与此处描述的系统和技术实施方式交互)、或者包括这种后台部件、中间件部件、或者前端部件的任何组合的计算系统中。可以通过任何形式或者介质的数

字数据通信(例如,通信网络)来将系统的部件相互连接。通信网络的示例包括:局域网(LAN)、广域网(WAN)和互联网。

[0055] 计算机系统可以包括客户端和服务端。客户端和服务端一般远离彼此并且通常通过通信网络进行交互。通过在相应的计算机上运行并且彼此具有客户端-服务端关系的计算机程序来产生客户端和服务端的关系。服务端可以是云服务器,也可以为分布式系统的服务端,或者是结合了区块链的服务端。

[0056] 应该理解,可以使用上面所示的各种形式的流程,重新排序、增加或删除步骤。例如,本公开中记载的各步骤可以并行地执行也可以顺序地执行也可以不同的次序执行,只要能够实现本公开公开的技术方案所期望的结果,本文在此不进行限制。

[0057] 上述具体实施方式,并不构成对本公开保护范围的限制。本领域技术人员应该明白的是,根据设计要求和因素,可以进行各种修改、组合、子组合和替代。任何在本公开的精神和原则之内所作的修改、等同替换和改进等,均应包含在本公开保护范围之内。

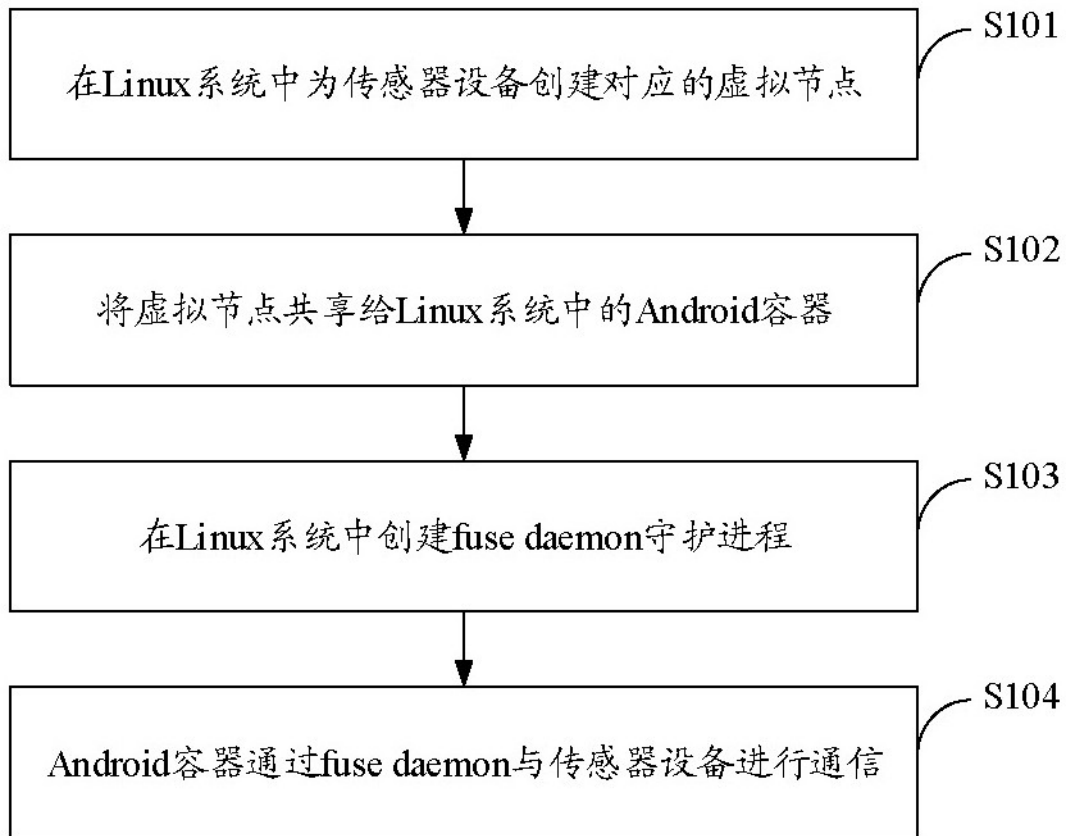


图1

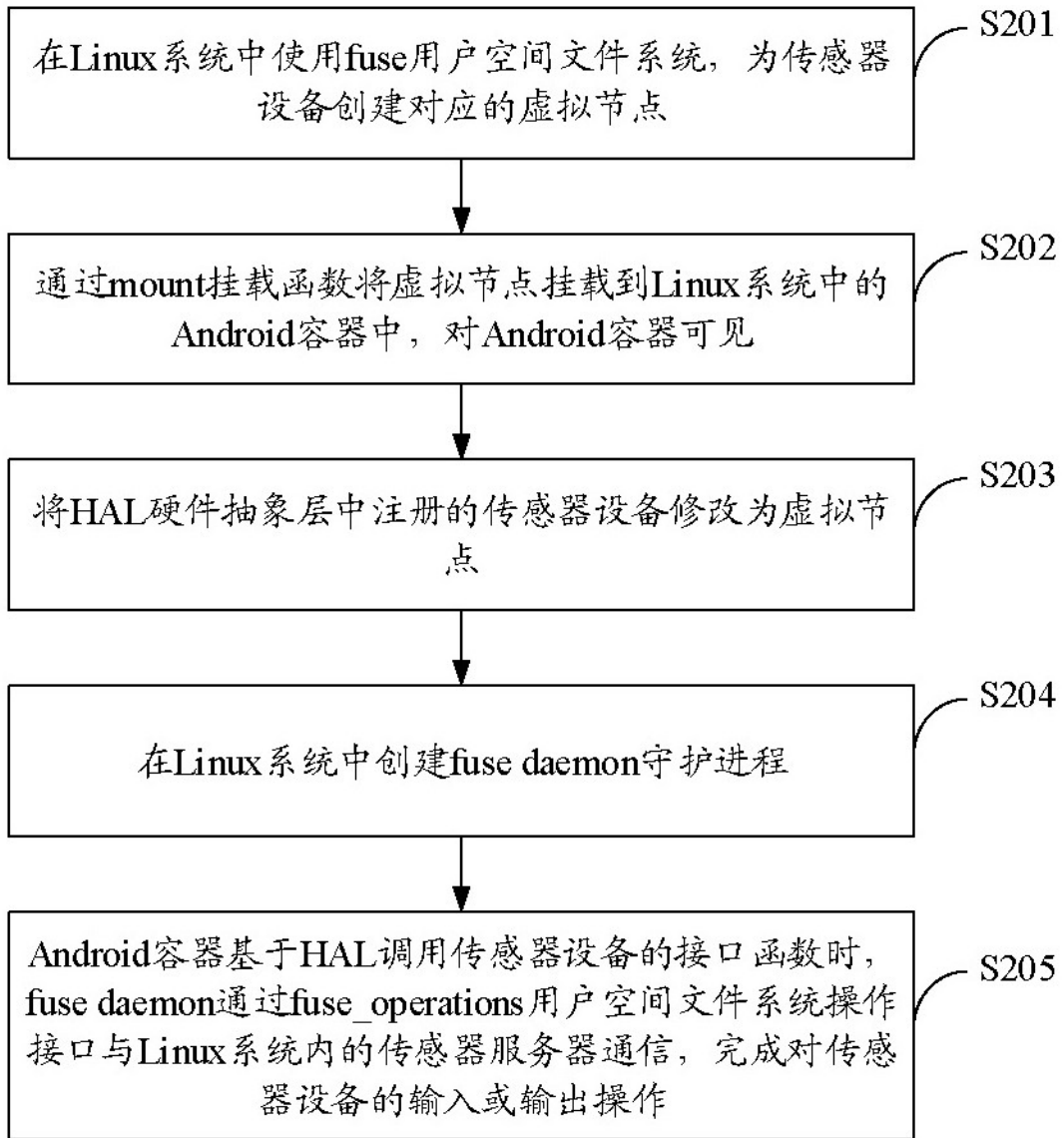


图2

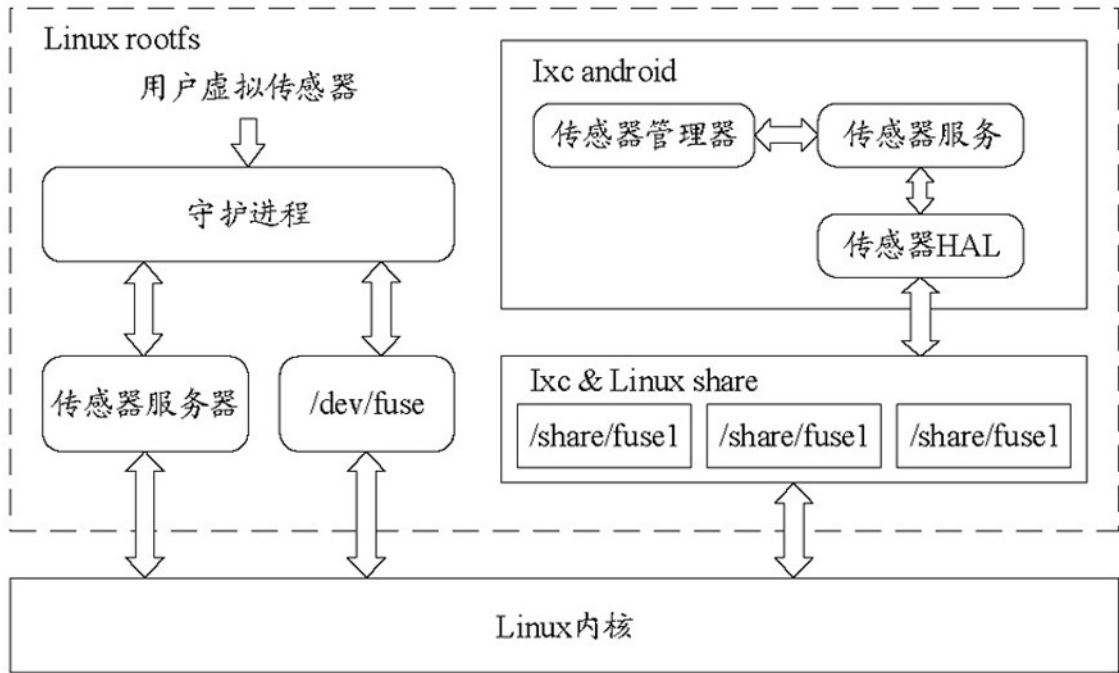


图3

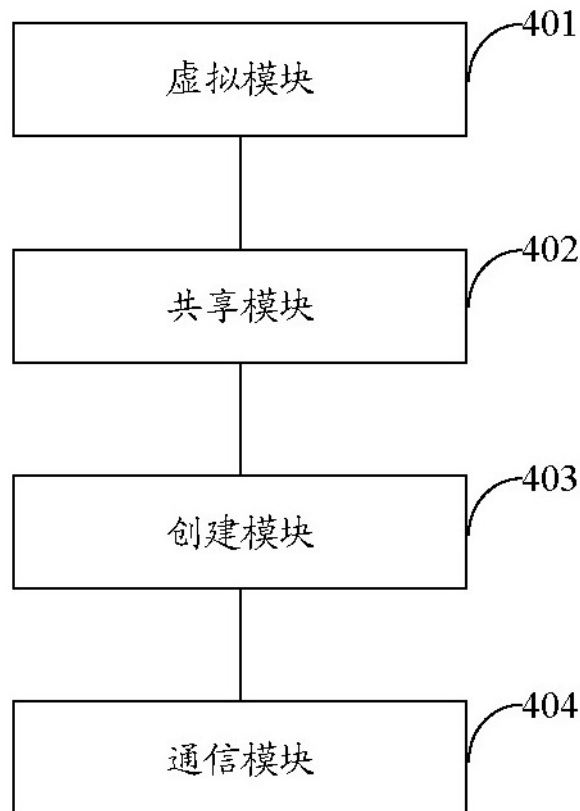


图4

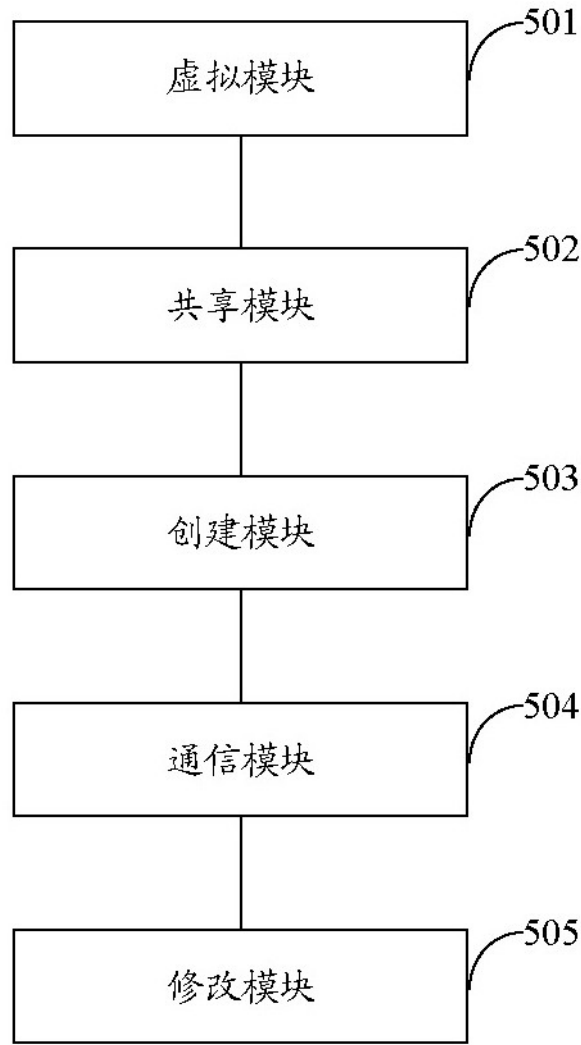


图5

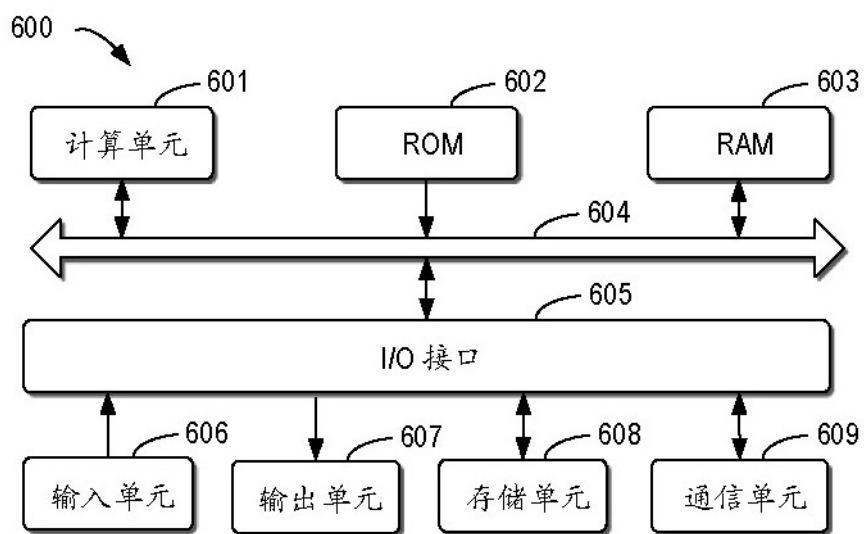


图6