



(12) 发明专利

(10) 授权公告号 CN 108595292 B

(45) 授权公告日 2021.03.26

(21) 申请号 201810383563.1

(22) 申请日 2018.04.26

(65) 同一申请的已公布的文献号  
申请公布号 CN 108595292 A

(43) 申请公布日 2018.09.28

(73) 专利权人 OPPO广东移动通信有限公司  
地址 523860 广东省东莞市长安镇乌沙海  
滨路18号

(72) 发明人 莫斯挺

(74) 专利代理机构 深圳中一联合知识产权代理  
有限公司 44414  
代理人 张全文

(51) Int.Cl.  
G06F 11/14 (2006.01)  
G06F 9/4401 (2018.01)

(56) 对比文件

US 8848576 B2, 2014.09.30

CN 106775876 A, 2017.05.31

佚名.APK安装流程详解8——

PackageManagerService的启动流程(下).  
《https://www.jianshu.com/p/b9b2b0f650e9》  
.2017, 第1-25页.

佚名.APK安装流程详解8——

PackageManagerService的启动流程(下).  
《https://www.jianshu.com/p/b9b2b0f650e9》  
.2017, 第1-25页.

洪双喜.Android平台隐私保护方法研究.  
《中国优秀博士论文全文数据库》.2018, 第85-86  
页.

审查员 刘凤娇

权利要求书2页 说明书9页 附图3页

(54) 发明名称

一种系统的优化方法、移动终端及计算机存  
储介质

(57) 摘要

本申请公开了一种系统的优化方法、移动终  
端及计算机存储介质,其中,所述方法包括:在系  
统启动之后,获取应用数据目录的唯一标识符  
UID;判断所述应用数据目录的UID是否异常;若  
所述应用数据目录的UID异常,则获取与所述应  
用数据目录关联的应用的UID,并将所述应用数  
据目录的UID替换为所述应用的UID。通过本申请  
可以在每次开机时自动对应用数据目录的UID进  
行修复,减小了用户启动应用时出现异常的概  
率,增强了系统的稳定性。



1. 一种系统的优化方法,其特征在于,  
在系统每次启动之后,通过installd进程获取应用数据目录的唯一标识符UID;  
判断所述应用数据目录的UID是否异常;  
若所述应用数据目录的UID异常,则获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID;  
其中,所述判断所述应用数据目录的UID是否异常包括:  
判断所述应用数据目录的UID是否与所述installd进程的UID相同;  
若所述应用数据目录的UID与所述installd进程的UID相同,则确定所述应用数据目录的UID异常。
2. 如权利要求1所述的系统的优化方法,其特征在于,所述获取与所述应用数据目录关联的应用的UID,包括:  
获取所述应用数据目录中的包名;  
基于所述包名获取与所述应用数据目录关联的应用的UID。
3. 如权利要求1所述的系统的优化方法,其特征在于,获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID包括:  
查找所述应用数据目录对应的应用安装目录;  
若查找到所述应用数据目录对应的应用安装目录,则从所述安装目录下获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID。
4. 如权利要求3所述的系统的优化方法,其特征在于,还包括:  
若未查找到所述应用数据目录对应的应用安装目录,则不获取与所述应用数据目录关联的应用的UID,并删除所述应用数据目录。
5. 如权利要求1至4任一项所述的系统的优化方法,其特征在于,在将所述应用数据目录的UID替换为所述应用的UID之后,还包括:  
启动所述应用;  
判断所述应用的启动是否发生异常;  
若所述应用的启动发生异常,则向用户发出提示信息或者重新启动所述系统。
6. 一种系统的优化装置,其特征在于,包括:  
UID获取单元,用于在系统每次启动之后,通过installd进程获取应用数据目录的唯一标识符UID;  
UID检测单元,用于判断所述应用数据目录的UID是否异常;  
UID修复单元,用于若所述应用数据目录的UID异常,则获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID;  
其中,所述判断所述应用数据目录的UID是否异常包括:  
判断所述应用数据目录的UID是否与所述installd进程的UID相同;  
若所述应用数据目录的UID与所述installd进程的UID相同,则确定所述应用数据目录的UID异常。
7. 一种移动终端,包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现如权利要求1至5任一项所述方法的步骤。

8. 一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如权利要求1至5任一项所述方法的步骤。

## 一种系统的优化方法、移动终端及计算机存储介质

### 技术领域

[0001] 本申请属于移动终端技术领域,尤其涉及一种系统的优化方法、移动终端及计算机存储介质。

### 背景技术

[0002] 安卓(Android)操作系统是目前移动终端上广泛使用的操作系统。在使用Android系统的手机时,经常会遇到如下问题:在安装应用app的时候,如果手机意外断电会导致应用发生异常,例如,当应用再次启动时,会提示无法找到该应用对应的应用数据目录。如果一旦出现这种情况,应用将无法操作自己的数据,导致功能异常。

### 发明内容

[0003] 有鉴于此,本申请提供了一种系统的优化方法、移动终端及计算机存储介质,以解决在应用安装过程中遇到断电所导致的应用无法操作应用数据的问题。

[0004] 本申请的第一方面提供了一种方法,上述方法包括:

[0005] 在系统启动之后,获取应用数据目录的唯一标识符UID;

[0006] 判断所述应用数据目录的UID是否异常;

[0007] 若所述应用数据目录的UID异常,则获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID。

[0008] 本申请的第二方面提供了一种装置,上述装置包括:

[0009] UID获取单元,用于在系统启动之后,获取应用数据目录的唯一标识符UID;

[0010] UID检测单元,用于判断所述应用数据目录的UID是否异常;

[0011] UID修复单元,用于若所述应用数据目录的UID异常,则获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID。

[0012] 本申请的第三方面提供了一种移动终端,上述移动终端包括存储器、处理器以及存储在上述存储器中并可在上述处理器上运行的计算机程序,上述处理器执行上述计算机程序时实现如上第一方面的方法的步骤。

[0013] 本申请的第二方面提供了一种计算机可读存储介质,上述计算机可读存储介质存储有计算机程序,上述计算机程序被处理器执行时实现如上第一方面的方法的步骤。

[0014] 本申请的第五方面提供了一种计算机程序产品,上述计算机程序产品包括计算机程序,上述计算机程序被一个或多个处理器执行时实现如上述第一方面的方法的步骤。

[0015] 由上可见,本申请方案中首先在系统启动之后,获取应用数据目录的唯一标识符UID,然后判断所述应用数据目录的UID是否异常,若所述应用数据目录的UID异常,则获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID。通过本申请的方案,在每次开机时自动对应用数据目录的UID进行检测修复,减小了用户启动应用时出现异常的概率,增强了系统的稳定性。

## 附图说明

[0016] 为了更清楚地说明本申请实施例中的技术方案,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申请的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0017] 图1是本申请实施例提供的一种系统的优化方法的实现流程示意图;

[0018] 图2是本申请实施例提供的一种系统的优化方法的另一实现流程示意图;

[0019] 图3是本申请实施例提供的一种系统的优化的另一实现流程示意图;

[0020] 图4是本申请实施例提供的一种系统的优化装置的结构框图;

[0021] 图5是本申请实施例提供的移动终端的示意图。

## 具体实施方式

[0022] 以下描述中,为了说明而不是为了限定,提出了诸如特定系统结构、技术之类的具体细节,以便透彻理解本申请实施例。然而,本领域的技术人员应当清楚,在没有这些具体细节的其它实施例中也可以实现本申请。在其它情况中,省略对众所周知的系统、装置、电路以及方法的详细说明,以免不必要的细节妨碍本申请的描述。

[0023] 为了说明本申请上述的技术方案,下面通过具体实施例来进行说明。

### [0024] 实施例一

[0025] 下面对本申请实施例提供的一种系统的优化方法方法进行描述,请参阅图1,本申请实施例中的方法包括:

[0026] 在步骤101中,在系统启动之后,获取应用数据目录的唯一标识符UID。

[0027] 在本申请实施例中,上述系统可以是安卓(Android)操作系统,或者是与Android操作系统类似的操作系统,该操作系统可以运行在智能手机、平板电脑、智能电视、路由器或者其他终端设备上。所述系统启动是指终端设备在接收到开机指令时进行上电操作,然后将操作系统加载到内存中运行,所述开机指令可以由用户通过操作终端设备上的开机键发出。

[0028] 在Android操作系统中,一个用户UID唯一标示一个应用。应用在安装时被分配唯一的UID,应用在设备上的存续期间内,UID保持不变。其中,每个应用对应一个应用数据目录,所述应用数据目录通常是系统上的文件目录/data/data/<包名>,应用数据目录用于存放应用的设置文件、数据库文件以及应用在运行中产生的数据等。例如终端设备上的微信应用,该应用程序被安装在/data/app/<微信对应的包名>文件目录下,微信应用在运行过程中产生的数据(包括临时数据和永久数据)被保存在/data/data/<微信对应的包名>文件目录。包名通常由应用提供商确定,例如微信的包名为“com.tencent.mm”。由于包名可以有不同的表现形式,在此不对包名的具体表现形式进行限定。在安装应用时,系统会创建应用数据目录,系统会给该应用数据目录分配与其关联的应用相同的UID。应用在运行的过程中,可以通过识别UID的方式识别用于存放该应用运行数据的应用数据目录。应用数据目录用于保存应用在运行时产生的数据。应用在运行过程中经常需要从应用数据目录中读取数据以及将应用运行时产生的数据存放在应用数据目录中。例如在使用微信聊天的过程中所产生的聊天记录就存放在微信应用的数据目录下,用户在向微信应用发出查看聊天记录的

指令时,微信应用通过读取与其关联的应用数据目录的相关数据,并将所述数据显示在供用户查看的UI界面上。

[0029] 在步骤102中,判断所述应用数据目录的UID是否异常。

[0030] 在步骤103中,若所述应用数据目录的UID异常,则获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID。

[0031] 在本申请实施例中,若应用数据目录的UID异常,例如,所述应用数据目录的UID不同于其关联应用的UID。由于应用是通过识别与自己具有相同UID的应用数据目录作为自己的应用数据目录。当应用数据目录的UID不同于其关联应用的UID时,该应用将无法查找到自己的应用数据目录,从而无法操作该应用关联的应用数据目录中的数据,此时认为应用数据目录的UID异常。并通过获取与该应用数据目录关联的应用的UID,然后将该应用数据目录的UID替换为该应用的UID的方式对应用数据目录进行修复。

[0032] 由于本申请实施例在系统每次启动之后,在应用程序启动之前,通过检测应用数据目录的UID是否出现异常,若出现异常则获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID,从而能够在每次开机时,修复异常的应用数据目录的UID,减小了用户启动应用时出现异常的概率,避免了在应用数据目录的UID异常未被修复之前启动关联的应用导致应用运行崩溃的问题,从而增强了系统的稳定性。

[0033] 应理解,上述实施例中各步骤的序号的大小并不意味着执行顺序的先后,各过程的执行顺序应以其功能和内在逻辑确定,而不应对本申请实施例的实施过程构成任何限定。

[0034] 实施例二

[0035] 图2是本申请实施例提供的另一种系统的优化方法的流程示意图,如图所示该方法可以包括以下步骤:

[0036] 在步骤201中,在系统启动之后,通过installd进程获取应用数据目录的UID。

[0037] 在本申请实施例中,installd是一个native的守护进程(daemon),installd具有root权限,能够直接管理包安装事务的事情。例如installd可以支持安装、卸载应用、删除apk、创建应用数据目录、更改应用数据目录的名称和应用数据目录的UID。installd通过监听本地的socket来执行相关命令,所述socket通过在init.rc文件中指定服务属性的方式创建。如果有socket连接进来,则通过socket读取命令字符串,然后执行命令。

[0038] 在步骤202中,判断所述应用数据目录的UID是否与所述installd进程的UID相同;

[0039] 在步骤203中,若所述应用数据目录的UID与所述installd进程的UID相同,则确定所述应用数据目录的UID异常。

[0040] 在本申请实施例中,installd在为应用创建应用数据目录时,首先会将正在创建的应用数据目录的UID设置为installd进程的UID,在应用数据目录创建完成之后,再将应用数据目录的UID修改为应用的UID。而在实践中发现,installd进程对应用进行操作的过程中,比如安装应用。当installd进程正在操作应用数据目录时(例如为应用创建应用数据目录)。如果发生突然掉电的事件。由于installd进程对应用数据目录的操作并没有完成,此时应用数据目录的UID实际上是installd进程的UID,而不是与应用数据目录关联的应用的UID。当再次开机启动系统之后,用户启动该应用时,该应用会通过UID查找对应的应用数据目录,由于与该应用关联的应用数据目录的UID是installd进程的UID,因此该应用由于

查找不到关联的应用数据目录而无法操作自己应用数据目录中的数据,从而导致应用运行异常。

[0041] 在步骤204中,若所述应用数据目录的UID异常,则获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID。

[0042] 在本申请实施例中,在系统启动之后,通过installld进程检测应用数据目录中是否存在UID与installld进程的UID相同的应用数据目录,如果存在UID与installld进程的UID相同的应用数据目录,则认为该应用数据目录的UID异常,应用程序将无法操作该应用数据目录。在本实施例中,可以通过获取与该应用数据目录关联的应用的UID,并将该应用数据目录的UID替换为该应用的UID,从而修复应用数据目录的UID异常的问题。

[0043] 由上可见,在本实施例中,在终端上电启动操作系统时,启动守护进程installld并通过isntallld进程对系统中所有应用数据目录的UID进行检测,若检测到存在应用数据目录的UID与installld进程的UID相同,则通过获取与该应用数据目录关联的应用的UID来替换应用数据目录的UID的方式进行UID修复。由于每次开机都使用installld进程对应用数据目录的UID进行检测,从而避免了意外断电导致的应用无法操作自己的数据目录的问题,增强了系统的稳定性。

[0044] 可选地,所述获取与所述应用数据目录关联的应用的UID,包括:

[0045] 获取所述应用数据目录中的包名;

[0046] 基于所述包名获取与所述应用数据目录关联的应用的UID。

[0047] 在本申请实施例中,当installld检测到UID异常的应用数据目录之后,可以通过该应用数据目录的路径信息获取应用的包名,例如,应用数据目录的路径为:/data/data/<包名>,通过截取该路径中最下一级目录的名称,并将截取的最下一级目录的名称赋值给字符串变量,即可获取应用数据目录的包名。

[0048] Android系统中的PackageMangerSerivce服务负责各种APK包的安装、卸载、优化和UID查询,可选地,本申请实施例可以通过包名信息向PackageMangerSerivce查询该包名所指代的应用的UID。在PackageMangerSerivce查询到应用的UID之后,将所述UID传递给installld进程,然后installld进程执行将查询到的应用的UID替换UID异常的应用数据目录的UID的操作,从而完成UID修复工作。

[0049] 本申请实施例通过installld获取UID异常的应用数据目录的包名,并基于所述包名向PackageMangerSerivce获取该应用的UID,通过installld与PackageMangerSerivce协同的方式完成开机后UID异常的修复工作,上述修复方式具有占用系统资源少,修复速度快,不影响开机启动效率等优点。而且在开机时对UID异常进行修复,避免用户遇到应用无法操作应用数据目录时,不知道如何解决的问题,提高了用户的体验。

[0050] 可选地,获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID包括:

[0051] 查找所述应用数据目录对应的应用安装目录;

[0052] 若查找到所述应用数据目录对应的应用安装目录,则从所述安装目录下获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID。

[0053] 其中,installld进程在获取到UID异常的应用数据目录的包名之后,可以在应用安装目录下(即/data/app路径)对该包名进行查找,如果查找到该包名,则表明该应用是已经

安装在该终端设备上的应用,通过在应用安装目录中查找应用的UID信息,并将查找到的应用的UID传递给installld进程,通过isntalld进程将查询到的应用的UID替换UID异常的应用数据目录的UID的操作,从而完成UID修复工作。

[0054] 可选地,若未查找到所述应用数据目录对应的应用安装目录,则不获取与所述应用数据目录关联的应用的UID,并删除所述应用数据目录。

[0055] 在本申请实施例中,installld进程在获取到UID异常的应用数据目录的包名之后,在应用安装目录下对该包名进行查找,若为查找到应用数据目录对应的应用安装目录,则表明该应用可能已经被卸载,由于该应用已经被卸载,也就是说不会出现启动该应用的事件,因此也就没有修复应用数据目录UID的必要,因此在为查找到应用数据目录对应的应用安装目录的情况下,不获取与应用数据目录关联的应用的UID。

[0056] 可选地,由于未查找到所述应用数据目录对应的应用安装目录表明该应用可能已经被卸载,该应用数据目录可能是在卸载应用的过程中意外断电而没有卸载完全的部分。因此在查找不到应用数据目录对应的应用安装目录时,删除该应用数据目录,从而节省终端的存储空间。

[0057] 应理解,上述实施例中各步骤的序号的大小并不意味着执行顺序的先后,各过程的执行顺序应以其功能和内在逻辑确定,而不应对本申请实施例的实施过程构成任何限定。

### [0058] 实施例三

[0059] 本实施例是以实施例一或实施例二为基础,图3是本申请实施例提供的另一种系统的优化方法的流程示意图,如图所示该方法在将所述应用数据目录的UID替换为所述应用的UID之后可以包括以下步骤:

[0060] 步骤301,启动所述应用;

[0061] 步骤302,判断所述应用的启动是否发生异常;

[0062] 步骤303,若所述应用的启动发生异常,则向用户发出提示信息或者重新启动所述系统。

[0063] 在本实施例中,在对UID异常的应用数据目录执行UID变更操作之后,可以自动执行启动应用的操作,例如,当系统检测到installld执行了将应用数据目录的UID进行替换的操作,则自动触发该应用的启动指令,所述启动方式可以是前台启动或者后台启动,优选采用后台启动的方式,在启动应用之后判断是否发生异常,所述异常例如可以是接收到应用的报错信息,或者应用自动退出。当检测到应用启动发生异常时,向用户发出提示信息。可选地,若在将所述应用数据目录的UID替换为所述应用的UID之后启动应用发生异常可以自动或者提示用户重启系统,并在重启系统之后,在此执行实施例一中的步骤101-103或者实施例二中的步骤201-204。在重启之后重新执行实施例一中的步骤101-103或者实施例二中的步骤201-204之后,如果启动应用依然存在异常,则向用户发出提示信息,建议用户采取其他解决方案,例如重装该应用或者对该应用进行升级操作。

[0064] 通过本申请实施例,在执行了UID修复操作之后,通过自动启动应用的方式检测所述修复是否成功,若检测到所述修复失败,则重启系统后重新执行修复操作。进一步提高了UID修复的成功率。

### [0065] 实施例四

[0066] 本申请实施例还提供一种系统的优化装置,如图3所示,本申请实施例中的优化装置400包括:

[0067] UID获取单元401,用于在系统启动之后,获取应用数据目录的唯一标识符UID;

[0068] UID检测单元402,用于判断所述应用数据目录的UID是否异常;

[0069] UID修复单元403,用于若所述应用数据目录的UID异常,则获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID。

[0070] 可选地,所述获取应用数据目录的UID包括:

[0071] 通过installd进程获取所述应用数据目录的UID。

[0072] 可选地,所述判断所述应用数据目录的UID是否异常包括:

[0073] 判断所述应用数据目录的UID是否与所述installd进程的UID相同;

[0074] 若所述应用数据目录的UID与所述installd进程的UID相同,则确定所述应用数据目录的UID异常。

[0075] 可选地,所述UID修复单元403,包括:

[0076] 查找子单元,用于查找所述应用数据目录对应的应用安装目录;

[0077] 替换子单元,用于若查找到所述应用数据目录对应的应用安装目录,则从所述安装目录下获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID。

[0078] 可选地,所述UID修复单元403还包括:

[0079] 删除子单元,用于若未查找到所述应用数据目录对应的应用安装目录,则不获取与所述应用数据目录关联的应用的UID,并删除所述应用数据目录。

[0080] 可选地,所述优化装置400还包括:

[0081] 检测单元,用于启动所述应用;

[0082] 判断所述应用的启动是否发生异常;

[0083] 若所述应用的启动发生异常,则向用户发出提示信息或者重新启动所述系统。

[0084] 使用本申请实施例中的系统的优化装置,由于本申请实施例在系统每次启动之后,在应用程序启动之前,通过检测应用数据目录的UID是否出现异常,若出现异常则获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID,从而能够在每次开机时,修复异常的应用数据目录的UID,避免了在应用数据目录的UID异常未被修复之前启动关联的应用导致应用运行崩溃的问题,减小了用户启动应用时出现异常的概率,增强了系统的稳定性。

[0085] 实施例五

[0086] 本申请实施例提供一种移动终端,请参阅图5,本申请实施例中的移动终端包括:存储器501,一个或多个处理器502(图5中仅示出一个)及存储在存储器501上并可在处理器上运行的计算机程序。其中:存储器501用于存储软件程序以及模块,处理器502通过运行存储在存储器401的软件程序以及单元,从而执行各种功能应用以及数据处理,以获取上述预设事件对应的资源。具体地,处理器502通过运行存储在存储器501的上述计算机程序时实现以下步骤:

[0087] 在系统启动之后,获取应用数据目录的唯一标识符UID;

[0088] 判断所述应用数据目录的UID是否异常;

[0089] 若所述应用数据目录的UID异常,则获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID。

[0090] 假设上述为第一种可能的实施方式,则在第一种可能的实施方式作为基础而提供的第二种可能的实施方式中,

[0091] 所述获取应用数据目录的UID包括:

[0092] 通过installd进程获取所述应用数据目录的UID。

[0093] 在上述第二种可能的实施方式作为基础而提供的第三种可能的实施方式中,

[0094] 所述判断所述应用数据目录的UID是否异常包括:

[0095] 判断所述应用数据目录的UID是否与所述installd进程的UID相同;

[0096] 若所述应用数据目录的UID与所述installd进程的UID相同,则确定所述应用数据目录的UID异常。

[0097] 在上述第一种可能的实施方式作为基础而提供的第四种可能的实施方式中,

[0098] 所述获取与所述应用数据目录关联的应用的UID,包括:

[0099] 获取所述应用数据目录中的包名;

[0100] 基于所述包名获取与所述应用数据目录关联的应用的UID。

[0101] 在上述第一种可能的实施方式作为基础而提供的第五种可能的实施方式中,

[0102] 获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID包括:

[0103] 查找所述应用数据目录对应的应用安装目录;

[0104] 若查找到所述应用数据目录对应的应用安装目录,则从所述安装目录下获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID;

[0105] 若未查找到所述应用数据目录对应的应用安装目录,则不获取与所述应用数据目录关联的应用的UID,并删除所述应用数据目录。

[0106] 在上述第一种可能的实施方式作为基础而提供的第七种可能的实施方式中,

[0107] 在将所述应用数据目录的UID替换为所述应用的UID之后,还包括:

[0108] 启动所述应用;

[0109] 判断所述应用的启动是否发生异常;

[0110] 若所述应用的启动发生异常,则向用户发出提示信息或者重新启动所述系统。

[0111] 进一步,如图5所示,上述移动终端5还可包括:一个或多个输入设备503(图5中仅示出一个)和一个或多个输出设备504(图5中仅示出一个)。存储器501、处理器502、输入设备503和输出设备404通过总线405连接。

[0112] 应当理解,上述移动终端5可以是智能手机、平板电脑等计算设备。所称处理器502可以是中央处理单元(Central Processing Unit,CPU),该处理器还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现成可编程门阵列(Field-Programmable Gate Array,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0113] 输入设备503可以包括键盘、触控板、指纹采传感器(用于采集用户的指纹信息和指纹的方向信息)、麦克风等,输出设备504可以包括屏幕、扬声器等。

[0114] 存储器501可以包括只读存储器和随机存取存储器,并向处理器502提供指令和数据。存储器501的一部分或全部还可以包括非易失性随机存取存储器。例如,存储器501还可以存储设备类型的信息。

[0115] 由上可见,在本申请实施例中,在应用程序启动之前,通过检测应用数据目录的UID是否出现异常,若出现异常则获取与所述应用数据目录关联的应用的UID,并将所述应用数据目录的UID替换为所述应用的UID,从而能够在每次开机时,修复异常的应用数据目录的UID,避免了在应用数据目录的UID异常未被修复之前启动关联的应用导致应用运行崩溃的问题,从而增强了系统的稳定性。

[0116] 所属领域的技术人员可以清楚地了解到,为了描述的方便和简洁,仅以上述各功能单元、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单元、模块完成,即将上述装置的内部结构划分成不同的功能单元或模块,以完成以上描述的全部或者部分功能。实施例中的各功能单元、模块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。另外,各功能单元、模块的具体名称也只是为了便于相互区分,并不用于限制本申请的保护范围。上述系统中单元、模块的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0117] 在上述实施例中,对各个实施例的描述都各有侧重,某个实施例中沒有详述或记载的部分,可以参见其它实施例的相关描述。

[0118] 本领域普通技术人员可以意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本申请的范围。

[0119] 在本申请所提供的实施例中,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,以上所描述的系统实施例仅仅是示意性的,例如,上述模块或单元的划分,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系统,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讯连接可以是通过一些接口,装置或单元的间接耦合或通讯连接,可以是电性,机械或其它的形式。

[0120] 上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单元来实现本实施例方案的目的。

[0121] 上述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,可以存储在一个计算机可读取存储介质中。基于这样的理解,本申请实现上述实施例方法中的全部或部分流程,也可以通过计算机程序来指令相关的硬件来完成,上述的计算机程序可存储于一计算机可读存储介质中,该计算机程序在被处理器执行时,可实现上述各个方法实施例的步骤。其中,上述计算机程序包括计算机程序代码,上述计算机程序代码可以为源代码形式、对象代码形式、可执行文件或某些中间形式等。上述计算机可读介质可以

包括:能够携带上述计算机程序代码的任何实体或装置、记录介质、U盘、移动硬盘、磁碟、光盘、计算机存储器、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、电载波信号、电信信号以及软件分发介质等。需要说明的是,上述计算机可读介质包含的内容可以根据司法管辖区内立法和专利实践的要求进行适当的增减,例如在某些司法管辖区,根据立法和专利实践,计算机可读介质不包括是电载波信号和电信信号。

[0122] 以上上述实施例仅用以说明本申请的技术方案,而非对其限制;尽管参照前述实施例对本申请进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围,均应包含在本申请的保护范围之内。

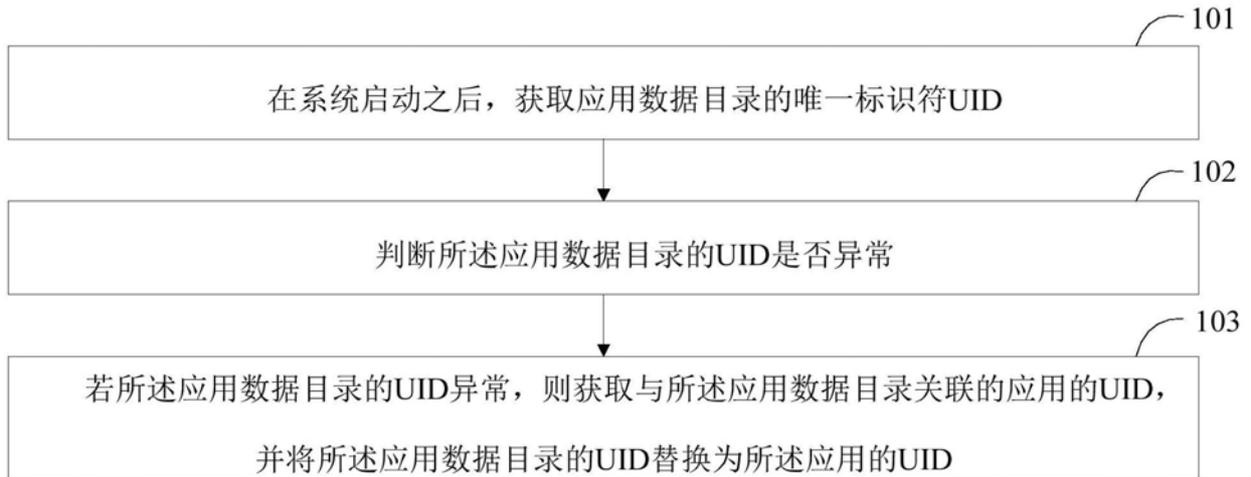


图1

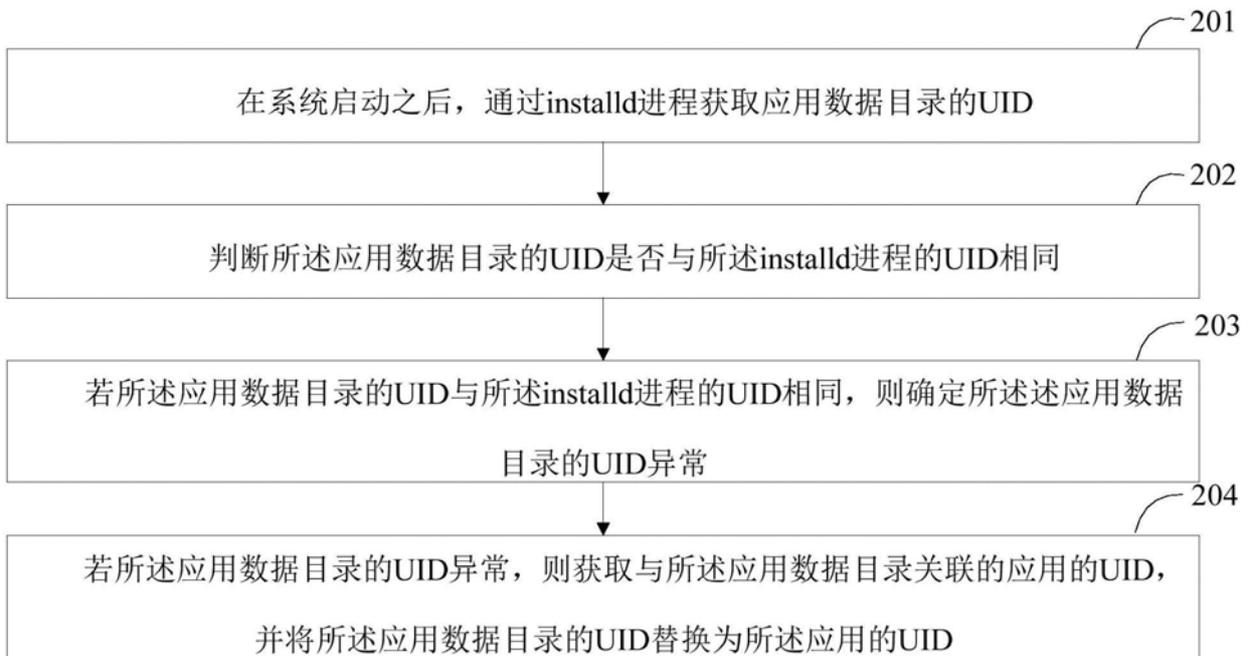


图2

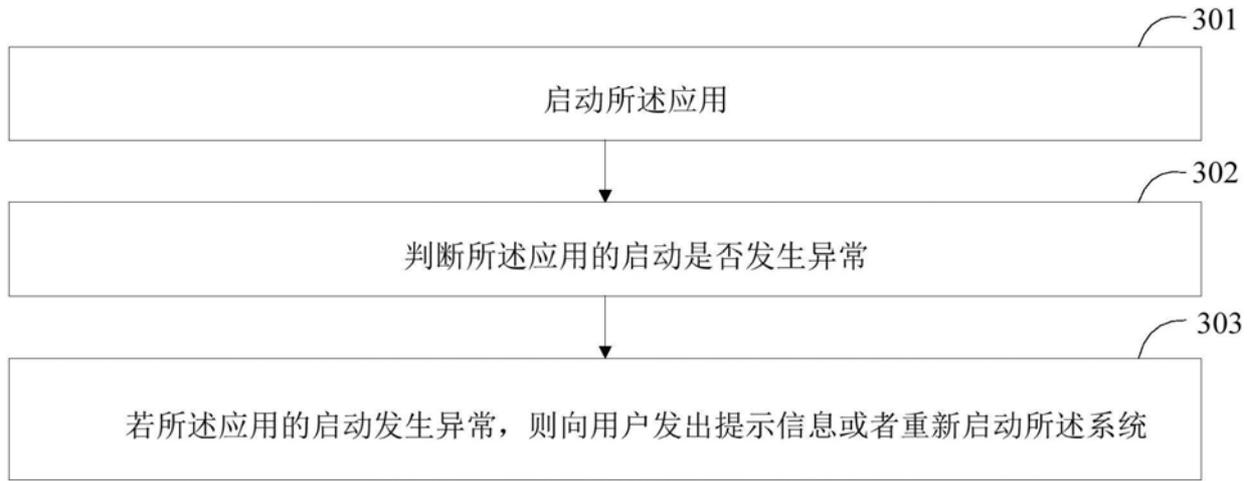


图3

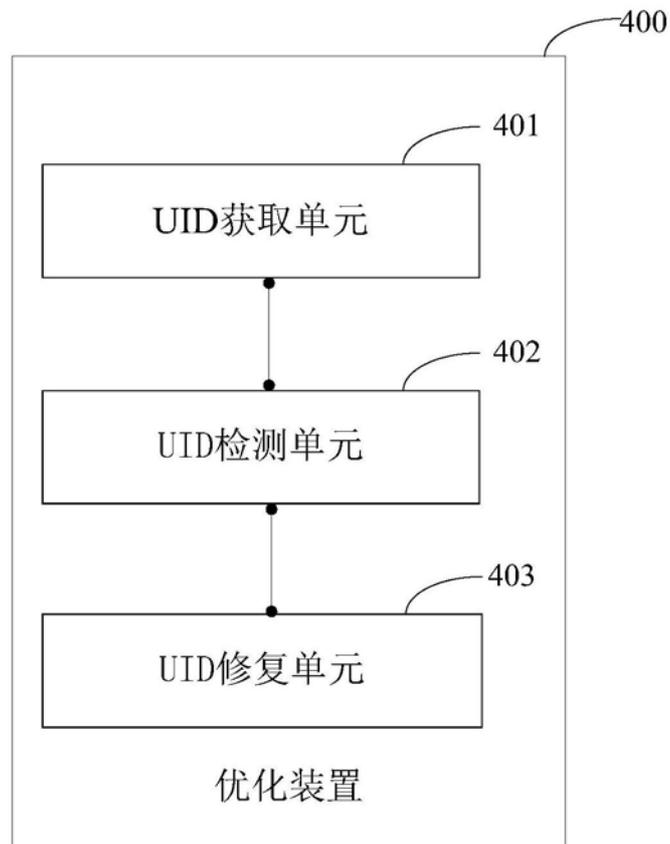


图4

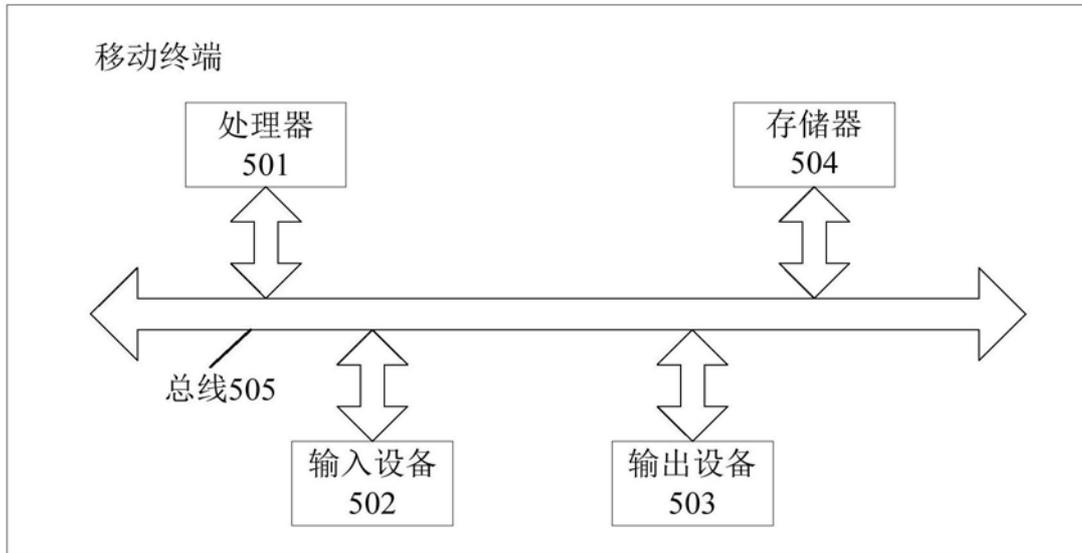


图5