



(12)发明专利申请

(10)申请公布号 CN 106569848 A

(43)申请公布日 2017. 04. 19

(21)申请号 201610900089.6

(22)申请日 2016.10.14

(71)申请人 惠州TCL移动通信有限公司
地址 516006 广东省惠州市仲恺高新区和畅七路西86号

(72)发明人 冯科 胡亚军 叶铨尧 王兵
赖一骏

(74)专利代理机构 深圳市威世博知识产权代理
事务所(普通合伙) 44280
代理人 何青瓦

(51)Int.Cl.
G06F 9/445(2006.01)

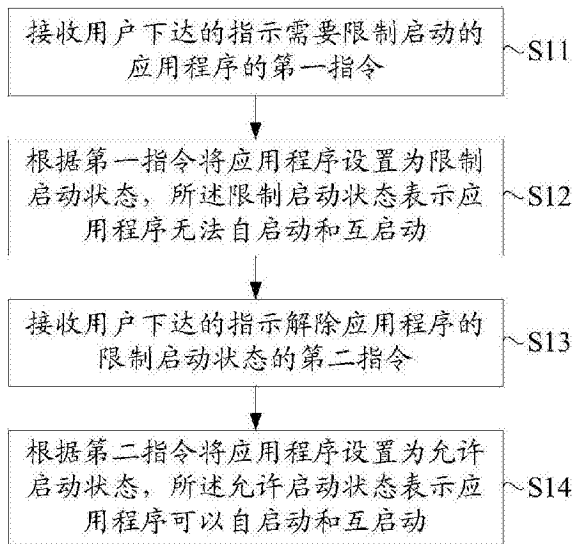
权利要求书2页 说明书7页 附图5页

(54)发明名称

终端及其应用程序的启动管理方法

(57)摘要

本发明公开一种终端及其应用程序的启动管理方法。该方法包括：根据用户下达的第一指令将应用程序设置为限制启动状态，所述限制启动状态表示应用程序无法自启动和互启动；根据用户下达的第二指令将应用程序设置为允许启动状态，所述允许启动状态表示应用程序可以自启动和互启动。本发明无需修改操作系统的平台代码即可限制应用程序的自启动和互启动，可移植性较强，且不会增加硬件成本。



1. 一种终端应用程序的启动管理方法,其特征在于,所述方法包括:
 - 接收用户下达的指示需要限制启动的应用程序的第一指令;
 - 根据所述第一指令将所述应用程序设置为限制启动状态,所述限制启动状态表示所述应用程序无法自启动和互启动;
 - 接收用户下达的指示解除应用程序的限制启动状态的第二指令;
 - 根据所述第二指令将所述应用程序设置为允许启动状态,所述允许启动状态表示所述应用程序可以自启动和互启动。
2. 根据权利要求1所述的方法,其特征在于,根据所述第二指令将所述应用程序设置为允许启动状态的步骤之后,所述方法还包括:
 - 检测所述应用程序是否处于使用状态;
 - 若所述应用程序未处于使用状态,则自动将所述应用程序设置为所述限制启动状态;
 - 若所述应用程序处于使用状态,则继续保持设置所述应用程序为所述允许启动状态。
3. 根据权利要求1所述的方法,其特征在于,
 - 所述根据所述第一指令将所述应用程序设置为限制启动状态的步骤,包括:
 - 将所述终端的操作系统中应用程序的静态广播组件设置为停用状态,使得所述应用程序无法自启动;
 - 将所述终端的操作系统中应用程序的服务组件和内容提供组件均设置为停用状态,使得所述应用程序无法互启动;
 - 所述根据所述第二指令将所述应用程序设置为允许启动状态的步骤,包括:
 - 将所述应用程序的静态广播组件设置为使用状态,使得所述应用程序可以自启动;
 - 将所述应用程序的服务组件和内容提供组件均设置为使用状态,使得所述应用程序可以互启动。
4. 根据权利要求3所述的方法,其特征在于,所述接收用户下达的指示解除应用程序的限制启动状态的第二指令的步骤之后,包括:
 - 检测操作系统中应用管理服务程序的监听进程是否切换至前台;
 - 在所述监听进程切换至前台时,确认用户下达了所述第二指令。
5. 根据权利要求3所述的方法,其特征在于,所述接收用户下达的指示解除应用程序的限制启动状态的第二指令的步骤之后,包括:
 - 检测操作系统中输入输出系统程序的事件信息中是否出现预定代码,所述预定代码为表示应用程序在终端的界面上被启动的代码;
 - 在出现所述预定代码时,确认用户下达了所述第二指令。
6. 一种终端,其特征在于,所述终端包括:
 - 接收模块,用于接收用户下达的指示需要限制启动的应用程序的第一指令;
 - 启动管理模块,用于根据所述第一指令将所述应用程序设置为限制启动状态,所述限制启动状态表示所述应用程序无法自启动和互启动;
 - 所述接收模块还用于接收用户下达的指示解除应用程序的限制启动状态的第二指令;
 - 所述启动管理模块还用于根据所述第二指令将所述应用程序设置为允许启动状态,所述允许启动状态表示所述应用程序可以自启动和互启动。
7. 根据权利要求6所述的终端,其特征在于,所述终端还包括检测模块,用于在所述启

动管理模块根据所述第二指令将所述应用程序设置为允许启动状态之后检测所述应用程序是否处于使用状态;若所述应用程序未处于使用状态,则所述启动管理模块还用于自动将所述应用程序设置为所述限制启动状态;若所述应用程序处于使用状态,则所述启动管理模块继续保持设置所述应用程序为所述允许启动状态。

8. 根据权利要求6所述的终端,其特征在于,

所述启动管理模块用于将所述终端的操作系统中应用程序的静态广播组件设置为停用状态,使得所述应用程序无法自启动,以及将所述操作系统中应用程序的服务组件和内容提供组件均设置为停用状态,使得所述应用程序无法互启动;

所述启动管理模块用于将所述应用程序的静态广播组件设置为使用状态,使得所述应用程序可以自启动,以及将所述应用程序的服务组件和内容提供组件均设置为使用状态,使得所述应用程序可以互启动。

9. 根据权利要求8所述的终端,其特征在于,所述检测模块还用于检测操作系统中应用管理服务程序的监听进程是否切换至前台,并在所述监听进程切换至前台时,确认用户下达了所述第二指令。

10. 根据权利要求8所述的终端,其特征在于,所述检测模块还用于检测操作系统中输入输出系统程序的事件信息中是否出现预定代码,所述预定代码为表示应用程序在终端的界面上被启动的代码,并在所述事件信息中出现所述预定代码时,确认用户下达了所述第二指令。

终端及其应用程序的启动管理方法

技术领域

[0001] 本发明涉及电子设备技术领域,具体涉及一种终端及其应用程序的启动管理方法。

背景技术

[0002] 随着电子技术的发展和人们生活水平的不断提高,手机等各种智能终端的使用越来越普及,人们经常会在终端上安装多种应用程序。但是作为第三方软件的应用程序,由于基于开发者的商业目的,其自身行为并未按照终端的操作系统规范进行约束,会在操作系统的后台偷偷启动,一旦启动的该类应用程序较多不仅会浪费网络流量,还容易造成终端内存占用率较大,极大的影响操作系统的运行速度。

[0003] 当前限制应用程序启动有两种方法:一种是修改操作系统的平台代码。虽然该方法可以较好的限制应用程序启动,但是平台代码的修改会导致操作系统的稳定性下降,维护难度大大增加,而且可移植性很低。另一种是限制使用操作系统的静态广播组件。但是该方法仅能限制应用程序的自启动,而无法限制应用程序之间的互启动。

发明内容

[0004] 有鉴于此,本发明实施例提供一种终端及其应用程序的启动管理方法,无需修改操作系统的平台代码即可限制应用程序的自启动和互启动,可移植性较强,且不会增加硬件成本。

[0005] 本发明一实施例的终端应用程序的启动管理方法,包括:接收用户下达的指示需要限制启动的应用程序的第一指令;根据第一指令将应用程序设置为限制启动状态,所述限制启动状态表示应用程序无法自启动和互启动;接收用户下达的指示解除应用程序的限制启动状态的第二指令;根据第二指令将应用程序设置为允许启动状态,所述允许启动状态表示应用程序可以自启动和互启动。

[0006] 可选地,所述根据第二指令将应用程序设置为允许启动状态的步骤之后,所述方法还包括:检测应用程序是否处于使用状态;若应用程序未处于使用状态,则自动将应用程序设置为限制启动状态;若应用程序处于使用状态,则继续保持设置应用程序为允许启动状态。

[0007] 可选地,所述根据第一指令将应用程序设置为限制启动状态的步骤,包括:将终端的操作系统中应用程序的静态广播组件设置为停用状态,使得应用程序无法自启动;将终端的操作系统中应用程序的服务组件和内容提供组件均设置为停用状态,使得应用程序无法互启动;所述根据第二指令将应用程序设置为允许启动状态的步骤,包括:将应用程序的静态广播组件设置为使用状态,使得应用程序可以自启动;将应用程序的服务组件和内容提供组件均设置为使用状态,使得应用程序可以互启动。

[0008] 可选地,所述接收用户下达的指示解除应用程序的限制启动状态的第二指令的步骤之后,包括:检测操作系统中应用管理服务程序的监听进程是否切换至前台;在监听进程

切换至前台时,确认用户下达了第二指令。

[0009] 可选地,所述接收用户下达的指示解除应用程序的限制启动状态的第二指令的步骤之后,包括:检测操作系统中输入输出系统程序的事件信息中是否出现预定代码,所述预定代码为表示应用程序在终端的界面上被启动的代码;在出现预定代码时,确认用户下达了第二指令。

[0010] 本发明一实施例的终端,包括:接收模块,用于接收用户下达的指示需要限制启动的应用程序的第一指令;启动管理模块,用于根据第一指令将应用程序设置为限制启动状态,所述限制启动状态表示应用程序无法自启动和互启动;接收模块还用于接收用户下达的指示解除应用程序的限制启动状态的第二指令;启动管理模块还用于根据第二指令将应用程序设置为允许启动状态,所述允许启动状态表示应用程序可以自启动和互启动。

[0011] 可选地,终端还包括检测模块,用于在启动管理模块根据第二指令将所述应用程序设置为允许启动状态之后检测应用程序是否处于使用状态;若应用程序未处于使用状态,则启动管理模块还用于自动将应用程序设置为限制启动状态;若应用程序处于使用状态,则启动管理模块继续保持设置所述应用程序为允许启动状态。

[0012] 可选地,启动管理模块用于将终端的操作系统中应用程序的静态广播组件设置为停用状态,使得应用程序无法自启动,以及将操作系统中应用程序的服务组件和内容提供组件均设置为停用状态,使得应用程序无法互启动;启动管理模块用于将应用程序的静态广播组件设置为使用状态,使得应用程序可以自启动,以及将应用程序的服务组件和内容提供组件均设置为使用状态,使得应用程序可以互启动。

[0013] 可选地,检测模块还用于检测操作系统中应用管理服务程序的监听进程是否切换至前台,并在监听进程切换至前台时,确认用户下达了第二指令。

[0014] 可选地,检测模块还用于检测操作系统中输入输出系统程序的事件信息中是否出现预定代码,所述预定代码为表示应用程序在终端的界面上被启动的代码,并在事件信息中出现预定代码时,确认用户下达了第二指令。

[0015] 有益效果:本发明实施例只有通过用户的主动操作才能允许应用程序的自启动和互启动,对应用程序的状态设置可以通过编程脚本或程序的方式实现,只需操作系统支持运行即可,无需改变操作系统的平台代码,可移植性较强,且不会增加硬件成本。

附图说明

[0016] 图1是本发明第一实施例的应用程序的启动管理方法的流程图;

[0017] 图2是本发明一实施例的限制应用程序启动的操作界面的示意图;

[0018] 图3是本发明一实施例的限制应用程序自启动的方法的流程图;

[0019] 图4是本发明一实施例的限制应用程序互启动的方法的流程图;

[0020] 图5是本发明一实施例的允许应用程序启动的方法的流程图;

[0021] 图6是本发明另一实施例的允许应用程序启动的方法的流程图;

[0022] 图7是本发明第二实施例的应用程序的启动管理方法的流程图;

[0023] 图8是本发明第三实施例的应用程序的启动管理方法的流程图;

[0024] 图9是本发明的终端一实施例的原理框示意图;

[0025] 图10是本发明的终端一实施例的硬件结构示意图。

具体实施方式

[0026] 下面将结合本发明实施例中的附图,对本发明所提供的各个示例性的实施例的技术方案进行清楚、完整地描述。在不冲突的情况下,下述各个实施例以及实施例中的特征可以相互组合。

[0027] 请参阅图1,为本发明一实施例的应用程序的启动管理方法。所述应用程序的启动管理方法可以包括步骤S11~S14。

[0028] S11:接收用户下达的指示需要限制启动的应用程序的第一指令。

[0029] S12:根据第一指令将应用程序设置为限制启动状态,所述限制启动状态表示应用程序无法自启动和互启动。

[0030] S13:接收用户下达的指示解除应用程序的限制启动状态的第二指令。

[0031] S14:根据第二指令将应用程序设置为允许启动状态,所述允许启动状态表示应用程序可以自启动和互启动。

[0032] 本发明实施例相当于为终端增加了一项控制应用程序启动与限制启动的功能。在现实应用场景中实现该功能的方式包括但不限于:以手机为例,首先,预先编程一脚本或程序,以在手机的设置界面中增加“限制启动应用程序”选项;然后,用户通过滑动滑块开启该选项,即可显示如图2所示的设置需要限制启动的应用程序的界面。其中,该界面显示终端安装的所有应用程序,并对每一应用程序显示一滑块,用户通过滑动滑块即可启动和限制启动对应的应用程序。

[0033] 也就是说,本实施例可以通过能够在操作系统上运行的脚本或程序的UI界面(例如滑动图2所示的滑块)下达第一指令和第二指令。相比较于现有技术的两种限制启动方式,本实施例可以通过编程脚本或程序的方式实现应用程序的限制启动,只需操作系统支持运行即可,无需改变操作系统的平台代码,可移植性较强,且不会增加硬件成本。

[0034] 本发明实施例的核心在于如何通过编程脚本或程序这一纯软件方式实现限制启动状态和允许启动状态的设置,即如何限制应用程序的启动以及允许应用程序的启动。

[0035] 对于限制应用程序的启动,本发明实施例可以采取禁用组件的方式,该方式可以在操作系统的应用层实现。以Android操作系统为例,参阅图3,限制应用程序自启动的方法可以包括步骤S31~S32;参阅图4,限制应用程序互启动的方法可以包括步骤S41~S42。

[0036] S31:通过操作系统的Package Manager(软件包管理)程序的get Package Info()函数(获取包信息函数)获取应用层信息;

[0037] S32:通过set Component Enabled Setting(组件激活)程序设置静态广播组件的状态为COMPONENT_ENABLED_STATE_DISABLED。

[0038] 也就是说,本实施例将操作系统中应用程序的静态广播组件设置为停用(Disable)状态,使得应用程序无法自启动。

[0039] S41:通过操作系统的Package Manager程序的get Package Info()函数获取应用层信息;

[0040] S42:通过set Component Enabled Setting的get Package Info()函数设置服务组件(Service)和内容提供组件(Provider)的状态为COMPONENT_ENABLED_STATE_DISABLED。

[0041] 也就是说,本实施例将操作系统中应用程序的服务组件和内容提供组件均设置为停用状态,使得应用程序无法互启动。

[0042] 对应地,本发明实施例可以将应用程序的静态广播组件设置为使用状态,使得应用程序可以自启动,并将应用程序的服务组件和内容提供组件均设置为使用状态,使得应用程序可以互启动。

[0043] 本发明实施例只有通过用户主动操作才能允许应用程序的启动,其关键在于终端如何检测到用户主动下达了前述第二指令,即如何监听用户点击终端界面上的图标以解除其启动限制这个行为。对此,本发明实施例可以采取如图5和图6所示的两种方法。

[0044] 参阅图5,该方法可以包括步骤S51~S53。

[0045] S51:检测操作系统中应用管理服务(Activity Manager Service,AMS)程序的监听进程(Process Observer)是否切换至前台。

[0046] 如果监听进程切换至前台,则执行步骤S52;如果未监听进程切换至前台,则执行步骤S53。

[0047] S52:确认用户下达了第二指令,而后将应用程序设置为允许启动状态。通过AMS/PMS和输入输出系统判断是否用户主动启动

[0048] S53:继续将应用程序设置为限制启动状态。

[0049] 其中,AMS检测到监听进程的状态为PROCESS_STATE_TOP,表示监听进程切换至前台,本实施例可以将该应用程序的静态广播组件、服务组件以及内容提供组件都设置为使用状态,并发送应用程序的状态改变的广播ACTION_PACKAGE_CHANGED。

[0050] 参阅图6,该方法可以包括步骤S61~S63。

[0051] S61:检测操作系统中输入输出系统程序的事件信息(event log)中是否出现预定代码,所述预定代码为表示应用程序在终端的界面上被启动的代码。

[0052] 如果出现预定代码,则执行步骤S62;如果未出现预定代码,则执行步骤S63。

[0053] S62:确认用户下达了第二指令,而后将应用程序设置为允许启动状态。

[0054] S63:继续将应用程序设置为限制启动状态。

[0055] 若检测事件信息包含代码am_create_activity或am_proc_start,表示启动了一个进程,若事件信息包含预定代码android.intent.action.MAIN,则表示该进程是在终端的界面上被启动的,本实施例可以将该应用程序的静态广播组件、服务组件以及内容提供组件都设置为使用状态。

[0056] 由上述可知,本发明实施例只有通过用户的主动操作才能允许应用程序的自启动和互启动,对应用程序的状态设置可以通过编程脚本或程序的方式实现,只需操作系统支持运行即可,无需改变操作系统的平台代码,可移植性较强,且不会增加硬件成本。并且,能够防止应用程序未按照操作系统规范的约束而在后台偷偷启动,不仅避免浪费网络流量,而且能够降低内存占用率,提高操作系统的运行速度。

[0057] 另外,由于内存占用率降低,本发明实施例还能增大终端的电源续航时间。例如,以手机安装运行20个应用程序为例,在手机没有限制应用程序的后台运行时,手机实时使用的电流数据是137.6毫安,对于电量为2000毫安时的电源,手机只能使用21.8个小时。而在限制应用程序的后台运行后,手机实时使用的电流数据是19.7毫安,那么对于具有同样电量的电池,手机可以使用152个小时,其耗电量与手机被打开飞行模式时的耗电量想接

近,大大提升了手机续航能力。

[0058] 请参阅图7,为本发明另一实施例的应用程序的启动管理方法。所述应用程序的启动管理方法可以包括步骤S71~S76。

[0059] S71:接收用户下达的指示需要限制启动的应用程序的第一指令。

[0060] S72:根据第一指令将应用程序设置为限制启动状态,所述限制启动状态表示应用程序无法自启动和互启动。

[0061] S73:接收用户下达的指示解除应用程序的限制启动状态的第二指令。

[0062] S74:根据第二指令将应用程序设置为允许启动状态,所述允许启动状态表示应用程序可以自启动和互启动。

[0063] S75:检测应用程序是否处于使用状态。

[0064] 例如,终端的屏幕由亮转灭或者终端被关机,均可以用于判定应用程序未处于使用状态。

[0065] 若应用程序未处于使用状态,则执行步骤S76。若应用程序处于使用状态,则继续保持设置应用程序为允许启动状态,并继续执行步骤S75。

[0066] S76:自动将应用程序设置为限制启动状态;

[0067] 在前述实施例的描述基础上,本发明实施例还可以在用户主动启动应用程序后,可以根据应用程序的实际使用状态决定是否再次限制应用程序的启动,使得未处于使用状态的应用程序再次被设置为限制启动状态,防止应用程序自启动和互启动。

[0068] 请参阅图8,为本发明的应用程序的启动管理方法一具体应用例。该实施例实现本发明前述实施例的整个过程,在此不作赘述。

[0069] 应该理解到,本发明实施例的上述功能如果以软件功能的形式实现并作为独立的产品销售或使用,可存储在一个计算机可读取存储介质中,即本发明实施例可以以软件产品的形式体现出来,其包括若干指令用以使得一台终端(可以是个人计算机、服务器或者网络设备等)执行本发明各实施例所述方法的全部或部分步骤。

[0070] 请参阅图9,为本发明一实施例的终端。所述终端80可以包括接收模块81和启动管理模块82。接收模块81用于接收用户下达的指示需要限制启动的应用程序的第一指令。启动管理模块82用于根据第一指令将应用程序设置为限制启动状态,所述限制启动状态表示应用程序无法自启动和互启动。接收模块81还用于接收用户下达的指示解除应用程序的限制启动状态的第二指令。启动管理模块82还用于根据第二指令将应用程序设置为允许启动状态,所述允许启动状态表示应用程序可以自启动和互启动。

[0071] 其中,启动管理模块82可以采取禁用组件的方式限制应用程序的启动。具体地,启动管理模块82将操作系统中应用程序的静态广播组件设置为停用状态,使得应用程序无法自启动,以及将操作系统中应用程序的服务组件和内容提供组件均设置为停用状态,使得应用程序无法互启动。对应地,启动管理模块82将应用程序的静态广播组件设置为使用状态,使得应用程序可以自启动,以及将应用程序的服务组件和内容提供组件均设置为使用状态,使得应用程序可以互启动。

[0072] 另外,启动管理模块82可以两种方式确认用户主动下达了所述第二指令,即监听用户点击终端80界面上的图标以解除其启动限制这个行为。一种方式是:终端80的检测模块83检测操作系统中应用管理服务程序的监听进程是否切换至前台,并在监听进程切换至

前台时,确认用户下达了第二指令。另一种方式是:检测模块83检测操作系统中输入输出系统程序的事件信息中是否出现预定代码,所述预定代码为表示应用程序在终端80的界面上被启动的代码,并在事件信息中出现预定代码时,确认用户下达了第二指令。

[0073] 本实施例的终端80还可以在用户主动启动应用程序后,根据应用程序的实际使用状态决定是否再次限制应用程序的启动,使得未处于使用状态的应用程序再次被设置为限制启动状态,防止应用程序自启动和互启动。具体地,检测模块83在启动管理模块82根据第二指令将应用程序设置为允许启动状态之后检测应用程序是否处于使用状态;若应用程序未处于使用状态,则启动管理模块82自动将应用程序设置为限制启动状态;若应用程序处于使用状态,则启动管理模块82继续保持设置所述应用程序为允许启动状态。

[0074] 本发明实施例的终端80的上述模块对应执行上述各个实施例的应用程序的启动管理方法,具有与其相同的技术效果。

[0075] 应该理解到,上述模块的划分为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如两个模块可以集成到另一个系统中,或一些特征可以忽略,或不执行。另外,模块相互之间的连接可以通过一些接口,也可以是电性或其它形式。上述模块既可以采用软件功能框的形式实现,也可以采用例如图10所示的硬件的形式实现。

[0076] 如图10所示,所述终端80包括至少一个处理器91、至少一个存储器92、至少一个通信接口93以及至少一个通信总线94,通信总线94以及通信接口93用于处理器91调用及获取各种数据,存储器92用于存储程序指令。为了方便说明,图10所示处理器91、存储器92、通信接口93以及通信总线94的数量为一个。

[0077] 其中,所述处理器91用于:

[0078] 执行程序指令以控制通信接口93接收用户下达的指示需要限制启动的应用程序的第一指令;

[0079] 执行程序指令以根据第一指令将应用程序设置为限制启动状态,所述限制启动状态表示应用程序无法自启动和互启动;

[0080] 执行程序指令以控制通信接口93接收用户下达的指示解除应用程序的限制启动状态的第二指令;

[0081] 执行程序指令以根据第二指令将应用程序设置为允许启动状态,所述允许启动状态表示应用程序可以自启动和互启动。

[0082] 其中,处理器91可以采取禁用组件的方式限制应用程序的启动。具体地,处理器91执行程序指令以将操作系统中应用程序的静态广播组件设置为停用状态,使得应用程序无法自启动,以及将操作系统中应用程序的服务组件和内容提供组件均设置为停用状态,使得应用程序无法互启动。对应地,处理器91执行程序指令以将应用程序的静态广播组件设置为使用状态,使得应用程序可以自启动,以及将应用程序的服务组件和内容提供组件均设置为使用状态,使得应用程序可以互启动。

[0083] 另外,处理器91可以两种方式确认用户主动下达了所述第二指令,即监听用户点击终端80界面上的图标以解除其启动限制这个行为。一种方式是:处理器91执行程序指令以检测操作系统中应用管理服务程序的监听进程是否切换至前台,并在监听进程切换至前台时,确认用户下达了第二指令。另一种方式是:处理器91执行程序指令以检测操作系统中输入输出系统程序的事件信息中是否出现预定代码,所述预定代码为表示应用程序在终端

80的界面上被启动的代码,并在事件信息中出现预定代码时,确认用户下达了第二指令。

[0084] 所述处理器91还可以在用户主动启动应用程序后,根据应用程序的实际使用状态决定是否再次限制应用程序的启动,使得未处于使用状态的应用程序再次被设置为限制启动状态,防止应用程序自启动和互启动。具体地,处理器91执行程序指令以在根据第二指令将应用程序设置为允许启动状态之后检测应用程序是否处于使用状态;若应用程序未处于使用状态,则自动将应用程序设置为限制启动状态;若应用程序处于使用状态,则继续保持设置所述应用程序为允许启动状态。

[0085] 在上述所有实施例中,所述终端可以为智能手机、PDA(Personal Digital Assistant,个人数字助理或平板电脑)、笔记本电脑等终端,也可以是佩戴于肢体或者嵌入于衣物、首饰、配件中的可穿戴设备,还可以是其他具有应用程序图标显示功能的智能电子设备。

[0086] 应理解,以上所述仅为本发明的实施例,并非因此限制本发明的专利范围,凡是利用本发明说明书及附图内容所作的等效结构或等效流程变换,例如各实施例之间技术特征的相互结合,或直接或间接运用在其他相关的技术领域,均同理包括在本发明的专利保护范围内。

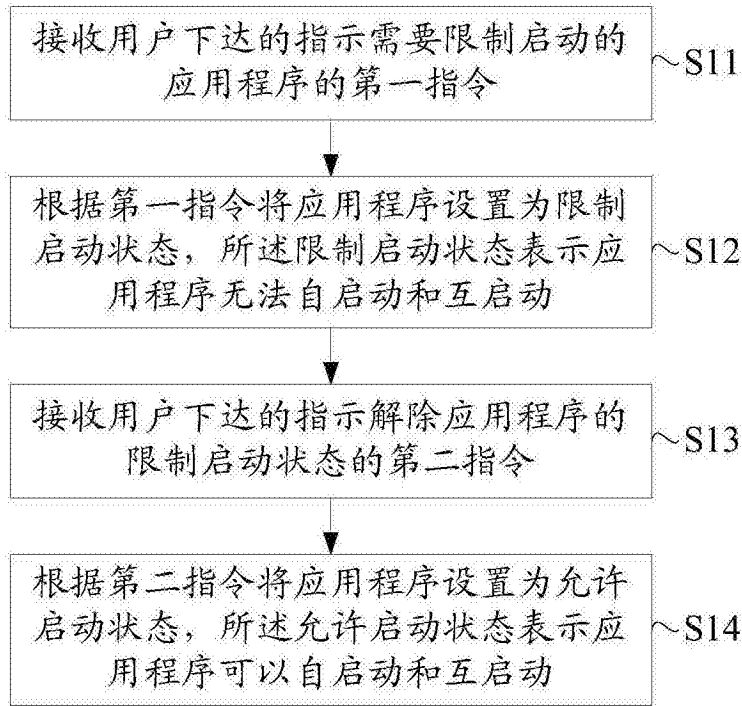


图1

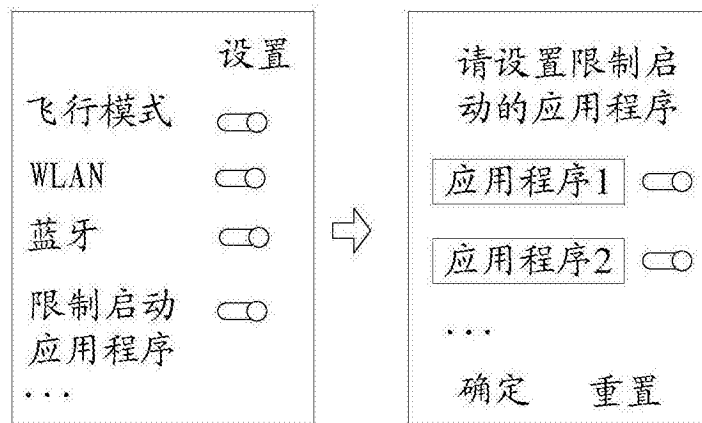


图2

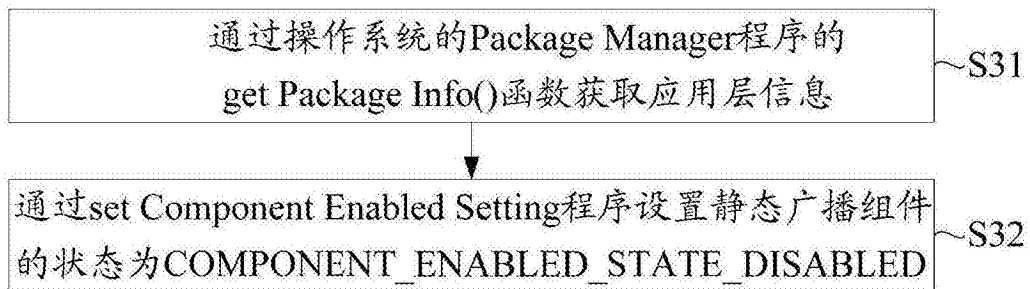


图3

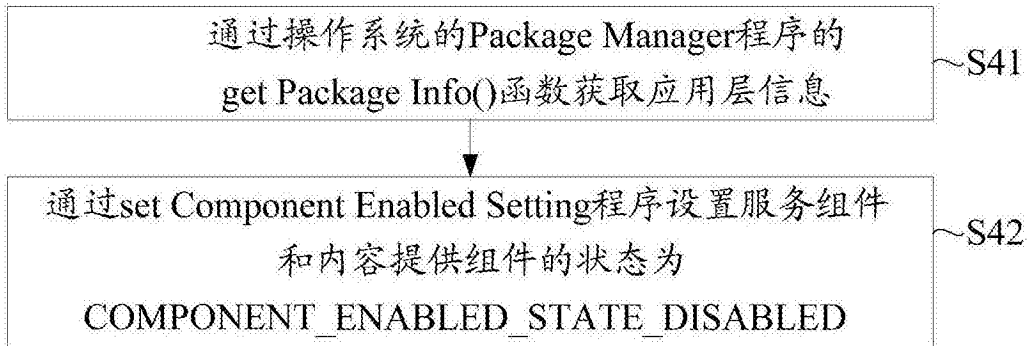


图4

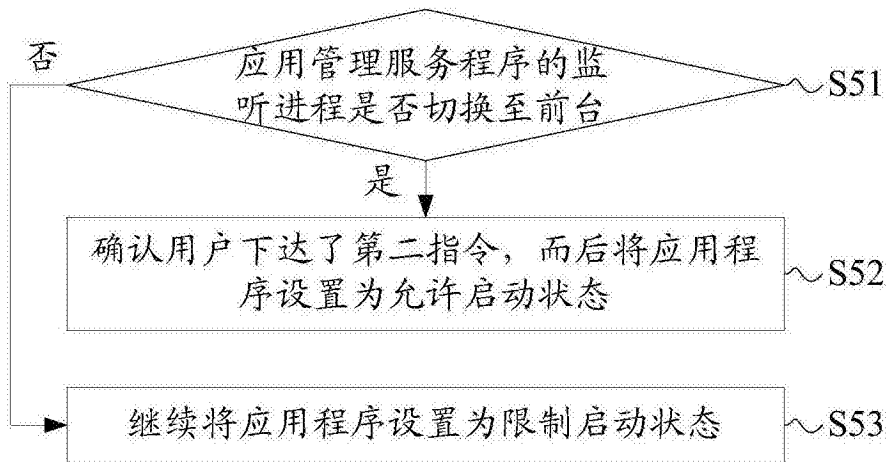


图5

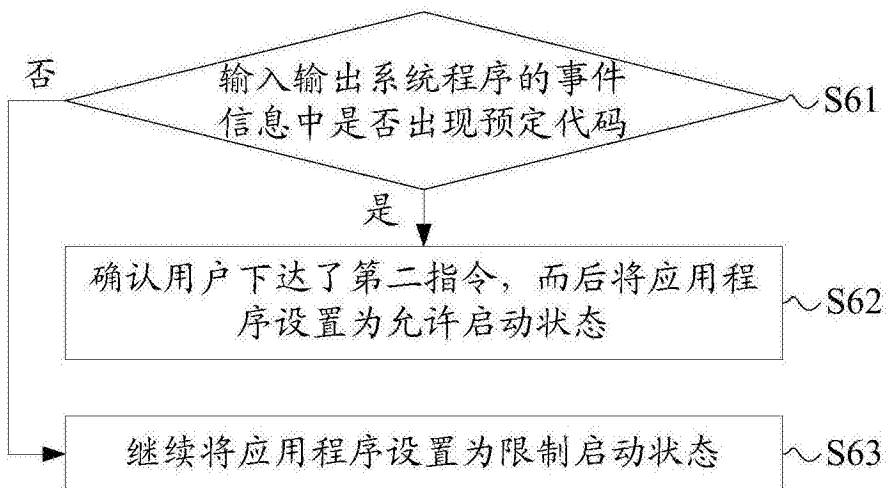


图6

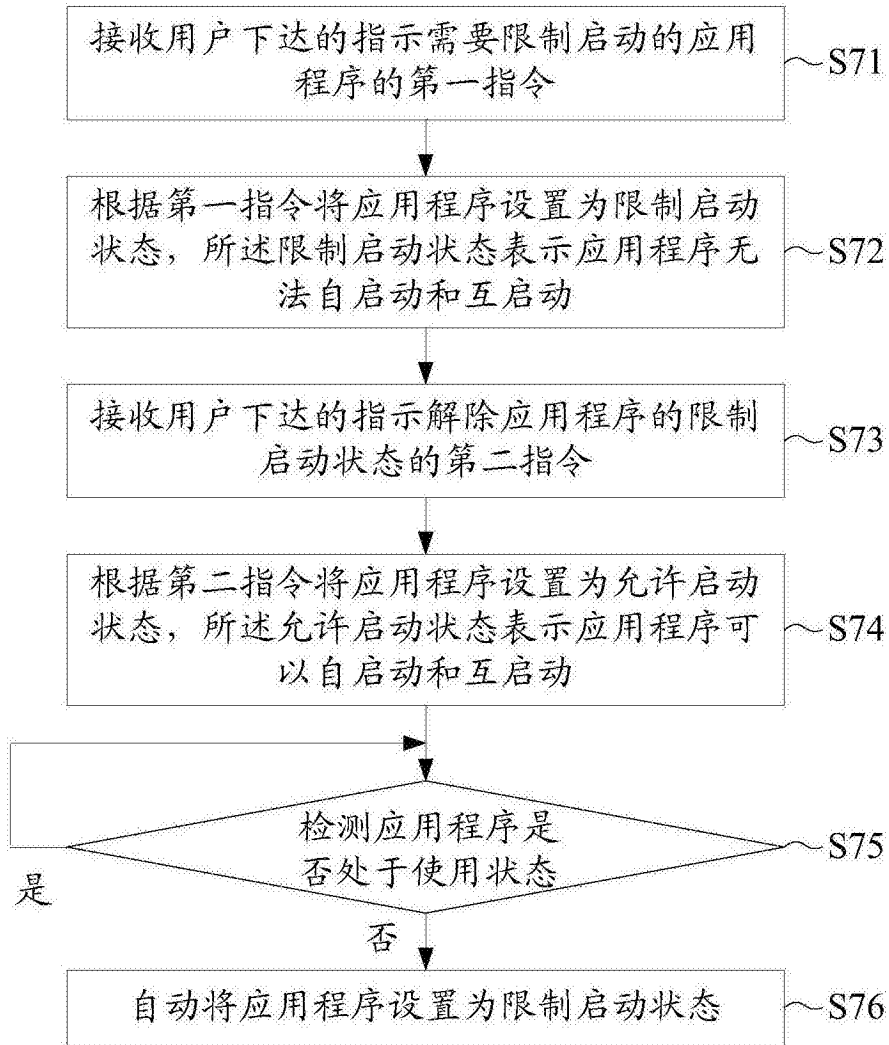


图7

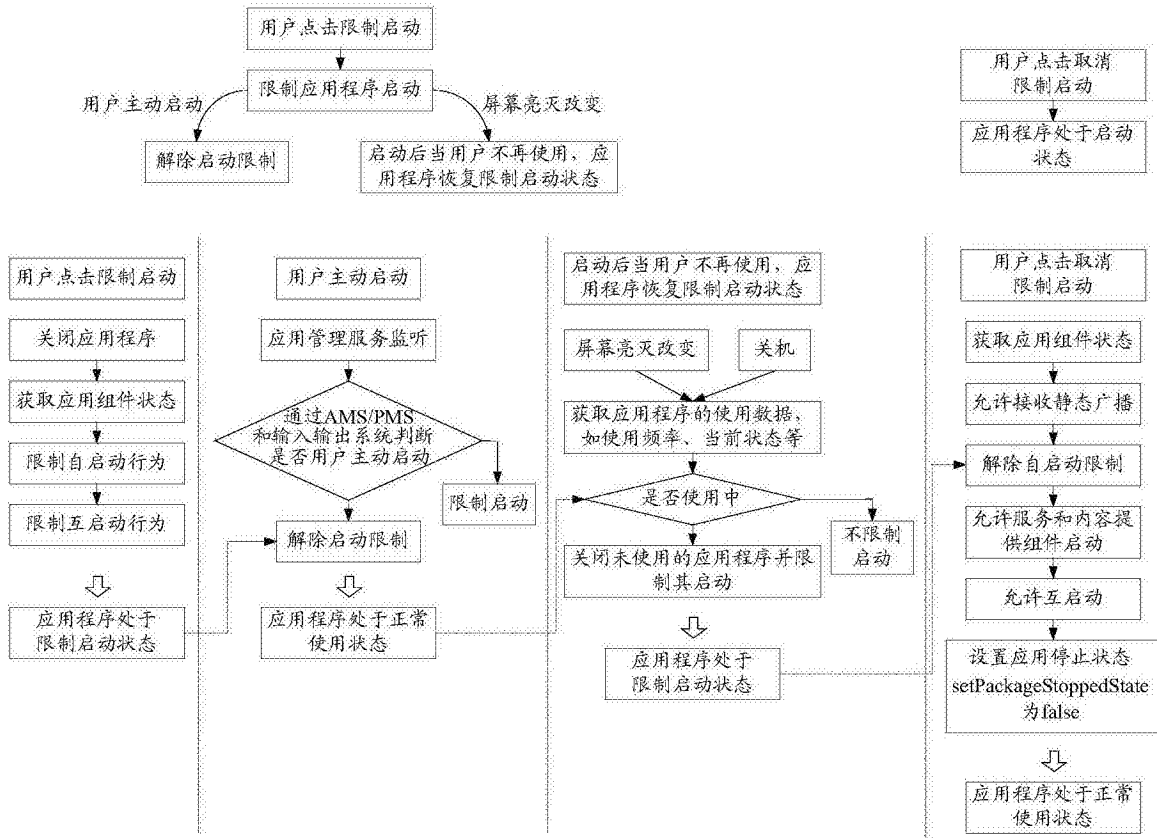


图8

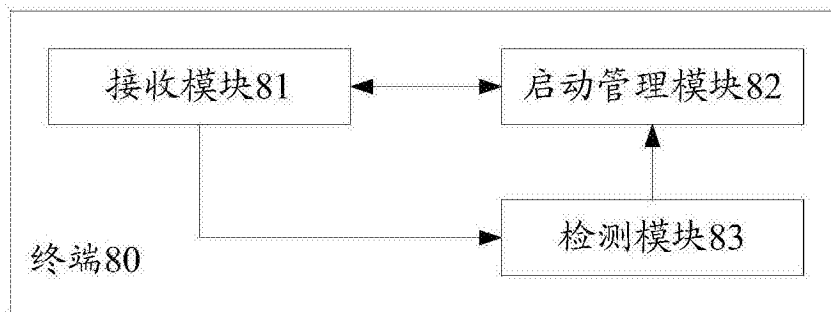


图9

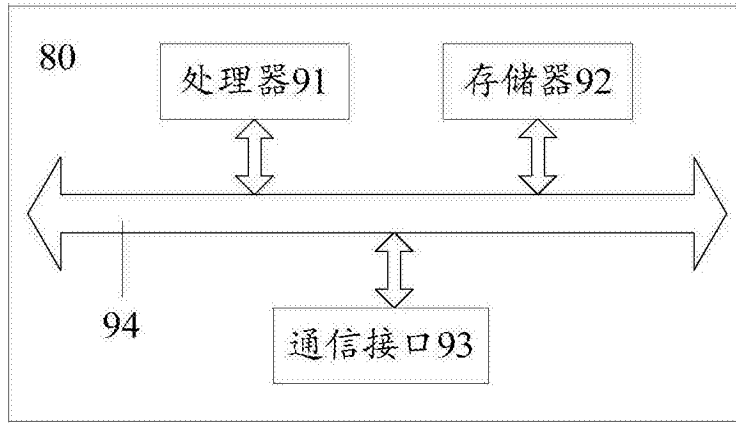


图10