



(12)发明专利申请

(10)申请公布号 CN 105955444 A

(43)申请公布日 2016.09.21

(21)申请号 201610260784.0

(22)申请日 2016.04.25

(71)申请人 深圳市万普拉斯科技有限公司  
地址 518000 广东省深圳市前海深港合作  
区前湾一路1号A栋201室

(72)发明人 刘惠盛

(74)专利代理机构 广州华进联合专利商标代理  
有限公司 44224

代理人 何平

(51) Int. Cl.  
G06F 1/32(2006.01)  
G06F 9/46(2006.01)

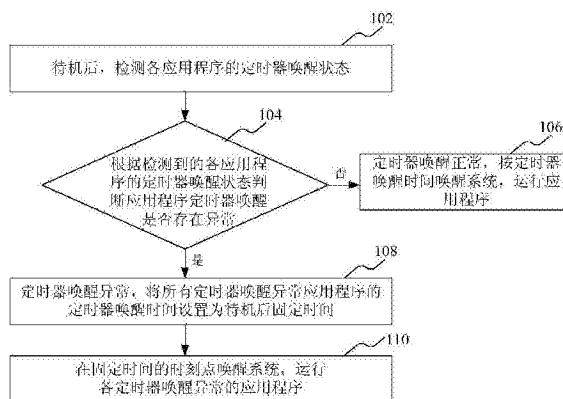
权利要求书2页 说明书7页 附图2页

(54)发明名称

对齐唤醒方法及装置

(57)摘要

本发明涉及一种对齐唤醒方法及装置,该对齐唤醒方法包括以下步骤:待机后,检测各应用程序的定时器唤醒状态;根据检测到的各应用程序的定时器唤醒状态判断应用程序定时器唤醒是否存在异常;若定时器唤醒正常,则按定时器唤醒时间唤醒系统,运行应用程序;若定时器唤醒异常,则将所有定时器唤醒异常应用程序的定时器唤醒时间设置为待机后固定时间;在所述固定时间的时刻点唤醒系统,运行各定时器唤醒异常的应用程序。该对齐唤醒方法既能够很好地节约终端待机耗电,达到省电的效果,又能够保证信息及时送达,实现了后台智能管理。此外,本发明还提出了一种对齐唤醒装置。



1. 一种对齐唤醒方法,其特征在于,包括以下步骤:  
待机后,检测各应用程序的定时器唤醒状态;  
根据检测到的各应用程序的定时器唤醒状态判断应用程序定时器唤醒是否存在异常;  
若定时器唤醒正常,则按定时器唤醒时间唤醒系统,运行应用程序;  
若定时器唤醒异常,则将所有定时器唤醒异常应用程序的定时器唤醒时间设置为待机后固定时间;  
在所述固定时间的时刻点唤醒系统,运行各定时器唤醒异常的应用程序。
2. 根据权利要求1所述的对齐唤醒方法,其特征在于,在所述检测各应用程序的定时器唤醒状态步骤中,所述定时器唤醒状态包括在预设时间内定时器唤醒系统的唤醒次数。
3. 根据权利要求2所述的对齐唤醒方法,其特征在于,所述根据检测到的各应用程序的定时器唤醒状态判断定时器唤醒是否存在异常的步骤包括以下步骤:  
判断应用程序的类型为定时唤醒程序或非定时唤醒程序;  
若应用程序为定时唤醒程序,则将检测到的预设时间内定时器唤醒系统的唤醒次数与预先设定的第一预设定时器唤醒次数进行比较,若预设时间内定时器的唤醒次数大于或等于所述第一预设定时器唤醒次数,则该应用程序为定时器唤醒异常应用程序;若预设时间内定时器的唤醒次数小于第一预设定时器唤醒次数,该应用程序为定时器唤醒正常应用程序;  
若应用程序为非定时唤醒程序,则将检测到的预设时间内定时器唤醒系统的唤醒次数与预先设定的第二预设定时器唤醒次数进行比较,若预设时间内定时器的唤醒次数大于或等于所述第二预设定时器唤醒次数,则所述应用程序为定时器唤醒异常应用程序;若预设时间内定时器的唤醒次数小于第二预设定时器唤醒次数,所述应用程序为定时器唤醒正常应用程序。
4. 根据权利要求1所述的对齐唤醒方法,其特征在于,在所述根据检测到的各应用程序的定时器唤醒状态判断应用程序定时器唤醒是否存在异常的步骤中包括以下步骤:若检测某应用程序的定时器唤醒状态失败,则所述应用程序为定时器唤醒异常应用程序。
5. 根据权利要求1所述的对齐唤醒方法,其特征在于,在所述的在所述固定时间时刻点唤醒系统,运行各定时器唤醒异常的应用程序的步骤之前还包括以下步骤:在上一次唤醒结束至下一次唤醒前的时间间隔内,通过后台进程接管各定时器唤醒异常应用程序的通知信息。
6. 一种对齐唤醒装置,其特征在于,包括:  
唤醒状态检测模块,用以检测各应用程序的定时器唤醒状态;  
唤醒异常判断模块,用以根据检测到的各应用程序的定时器唤醒状态判断应用程序定时器唤醒是否存在异常;  
对齐唤醒模块,用以将所有定时器唤醒异常应用程序的定时器唤醒时间设置为待机后固定时间;以及,  
唤醒触发模块,用以唤醒系统,运行应用程序。
7. 根据权利要求6所述的对齐唤醒装置,其特征在于,所述唤醒状态检测模块用以检测预设时间内定时器唤醒系统的唤醒次数。
8. 根据权利要求7所述的对齐唤醒装置,其特征在于,所述唤醒异常判断模块包括:

应用程序类型判断子模块,用以判断应用程序的类型为定时唤醒程序或非定时唤醒程序;

存储子模块,用以存储预设时间、第一预设定时器唤醒次数和第二预设定时器唤醒次数;

第一唤醒异常判断子模块,用以判断定时唤醒程序的定时器唤醒是否存在异常,若应用程序为定时唤醒程序,所述第一唤醒异常判断子模块将检测到的预设时间内定时器唤醒系统的唤醒次数与的第一预设定时器唤醒次数进行比较,若预设时间内定时器的唤醒次数大于或等于所述第一预设定时器唤醒次数,则该应用程序为定时器唤醒异常应用程序;若预设时间内定时器的唤醒次数小于第一预设定时器唤醒次数,该应用程序为定时器唤醒正常应用程序;

第二唤醒异常判断子模块,用以判断非定时唤醒程序的定时器唤醒是否存在异常,若应用程序为非定时唤醒程序,所述第二唤醒异常判断子模块将检测到的预设时间内定时器唤醒系统的唤醒次数与第二预设定时器唤醒次数进行比较,若预设时间内定时器的唤醒次数大于或等于所述第二预设定时器唤醒次数,则所述应用程序为定时器唤醒异常应用程序;若预设时间内定时器的唤醒次数小于第二预设定时器唤醒次数,所述应用程序为定时器唤醒正常应用程序。

9. 根据权利要求6所述的对齐唤醒装置,其特征在于,所述唤醒异常判断模块包括:

唤醒状态统计子模块,用以检测并统计预设时间内定时器唤醒系统的唤醒状态,若检测应用程序的定时器唤醒状态成功,则所述唤醒状态检测子模块统计预设时间内定时器唤醒系统的唤醒次数;若检测某应用程序的定时器唤醒状态失败,则所述唤醒状态检测子模块定义该应用程序为定时器唤醒异常应用程序。

10. 根据权利要求6所述的对齐唤醒装置,其特征在于,所述对齐唤醒系统还包括通知信息管理模块,用以在上一次唤醒结束至下一次唤醒前的时间间隔内,通过后台进程接管各定时器唤醒异常应用程序的通知信息。

## 对齐唤醒方法及装置

### 技术领域

[0001] 本发明涉及移动终端技术领域,特别是涉及一种对齐唤醒方法及装置。

### 背景技术

[0002] 随着终端技术的发展,各种智能操作系统都提供了定时器唤醒机制。为了避免不同的应用繁杂无序的唤醒终端系统,增加终端的待机功耗,近年来提出了对齐唤醒机制。目前,普遍采用的对齐唤醒方案都是延迟各个应用程序的定时器唤醒时间,然后集中所有定时器到一个时间点来触发终端系统,通过调整各应用程序的定时器唤醒时间,使各应用程序的定时器对齐唤醒终端系统,以达到待机时省电的效果。然而,该种方案会误导致一些正常的应用程序定时器不能准时唤醒系统,尤其是对于一些在待机后设置较少定时器唤醒次数的应用程序不能正常触发。

### 发明内容

[0003] 基于此,有必要针对传统对齐唤醒机制存在的不能准时唤醒正常定时器的的问题,提供一种对齐唤醒方法及装置。

[0004] 一种对齐唤醒方法,包括以下步骤:

[0005] 待机后,检测各应用程序的定时器唤醒状态;

[0006] 根据检测到的各应用程序的定时器唤醒状态判断应用程序定时器唤醒是否存在异常;

[0007] 若定时器唤醒正常,则按定时器唤醒时间唤醒系统,运行应用程序;

[0008] 若定时器唤醒异常,则将所有定时器唤醒异常应用程序的定时器唤醒时间设置为待机后固定时间;

[0009] 在所述固定时间的时刻点唤醒系统,运行各定时器唤醒异常的应用程序。

[0010] 在其中一个实施例中,在所述检测各应用程序的定时器唤醒状态步骤中,所述定时器唤醒状态包括在预设时间内定时器唤醒系统的唤醒次数。

[0011] 在其中一个实施例中,所述根据检测到的各应用程序的定时器唤醒状态判断定时器唤醒是否存在异常的步骤包括以下步骤:

[0012] 判断应用程序的类型为定时唤醒程序或非定时唤醒程序;

[0013] 若应用程序为定时唤醒程序,则将检测到的预设时间内定时器唤醒系统的唤醒次数与预先设定的第一预设定时器唤醒次数进行比较,若预设时间内定时器的唤醒次数大于或等于所述第一预设定时器唤醒次数,则该应用程序为定时器唤醒异常应用程序;若预设时间内定时器的唤醒次数小于第一预设定时器唤醒次数,该应用程序为定时器唤醒正常应用程序;

[0014] 若应用程序为非定时唤醒程序,则将检测到的预设时间内定时器唤醒系统的唤醒次数与预先设定的第二预设定时器唤醒次数进行比较,若预设时间内定时器的唤醒次数大于或等于所述第二预设定时器唤醒次数,则所述应用程序为定时器唤醒异常应用程序;若

预设时间内定时器的唤醒次数小于第二预设定时器唤醒次数,所述应用程序为定时器唤醒正常应用程序。

[0015] 在其中一个实施例中,在所述根据检测到的各应用程序的定时器唤醒状态判断应用程序定时器唤醒是否存在异常的步骤中包括以下步骤:若检测某应用程序的定时器唤醒状态失败,则所述应用程序为定时器唤醒异常应用程序。

[0016] 在其中一个实施例中,在所述的在所述固定时间时刻点唤醒系统,运行各定时器唤醒异常的应用程序的步骤之前还包括以下步骤:在上一次唤醒结束至下一次唤醒前的时间间隔内,通过后台进程接管各定时器唤醒异常应用程序的通知信息。

[0017] 一种对齐唤醒装置,包括:

[0018] 唤醒状态检测模块,用以检测各应用程序的定时器唤醒状态;

[0019] 唤醒异常判断模块,用以根据检测到的各应用程序的定时器唤醒状态判断应用程序定时器唤醒是否存在异常;

[0020] 对齐唤醒模块,用以将所有定时器唤醒异常应用程序的定时器唤醒时间设置为待机后固定时间;以及,

[0021] 唤醒触发模块,用以唤醒系统,运行应用程序。

[0022] 在其中一个实施例中,所述唤醒状态检测模块用以检测预设时间内定时器唤醒系统的唤醒次数。

[0023] 在其中一个实施例中,所述唤醒异常判断模块包括:

[0024] 应用程序类型判断子模块,用以判断应用程序的类型为定时唤醒程序或非定时唤醒程序;

[0025] 存储子模块,用以存储预设时间、第一预设定时器唤醒次数和第二预设定时器唤醒次数;

[0026] 第一唤醒异常判断子模块,用以判断定时唤醒程序的定时器唤醒是否存在异常,若应用程序为定时唤醒程序,所述第一唤醒异常判断子模块将检测到的预设时间内定时器唤醒系统的唤醒次数与的第一预设定时器唤醒次数进行比较,若预设时间内定时器的唤醒次数大于或等于所述第一预设定时器唤醒次数,则该应用程序为定时器唤醒异常应用程序;若预设时间内定时器的唤醒次数小于第一预设定时器唤醒次数,该应用程序为定时器唤醒正常应用程序;

[0027] 第二唤醒异常判断子模块,用以判断非定时唤醒程序的定时器唤醒是否存在异常,若应用程序为非定时唤醒程序,所述第二唤醒异常判断子模块将检测到的预设时间内定时器唤醒系统的唤醒次数与第二预设定时器唤醒次数进行比较,若预设时间内定时器的唤醒次数大于或等于所述第二预设定时器唤醒次数,则所述应用程序为定时器唤醒异常应用程序;若预设时间内定时器的唤醒次数小于第二预设定时器唤醒次数,所述应用程序为定时器唤醒正常应用程序。

[0028] 在其中一个实施例中,所述唤醒异常判断模块包括:

[0029] 唤醒状态统计子模块,用以检测并统计预设时间内定时器唤醒系统的唤醒状态,若检测应用程序的定时器唤醒状态成功,则所述唤醒状态检测子模块统计预设时间内定时器唤醒系统的唤醒次数;若检测某应用程序的定时器唤醒状态失败,则所述唤醒状态检测子模块定义该应用程序为定时器唤醒异常应用程序。

[0030] 在其中一个实施例中,所述对齐唤醒系统还包括通知信息管理模块,用以在上一次唤醒结束至下一次唤醒前的时间间隔内,通过后台进程接管各定时器唤醒异常应用程序的通知信息。

[0031] 上述的对齐唤醒方法及装置,对设置过多定时器唤醒次数的应用程序做唤醒延迟处理,对正常定时器唤醒的应用程序不做延时处理,实现了对各应用程序的定时器唤醒次数的针对性管理,既保证了设置正常的应用程序能够准时唤醒系统,确保信息及时送达,又实现了对设置频繁唤醒次数的应用程序对齐唤醒,能够很好地节约终端待机耗电,达到省电的效果,实现后台智能管理。

### 附图说明

[0032] 图1为一个实施例中对齐唤醒方法的流程原理图;

[0033] 图2为一个实施例中判断定时器唤醒是否存在异常的流程原理图;

[0034] 图3为一个实施例中对齐唤醒装置的结构原理图;

[0035] 图4为一个实施例中唤醒异常判断模块的结构原理图。

### 具体实施方式

[0036] 为了使本发明的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本发明进行进一步详细说明。应当理解,此处所描述的具体实施例仅仅用以解释本发明,并不用于限定本发明。

[0037] 请参阅图1,一种对齐唤醒方法及系统,包括以下步骤:

[0038] 步骤102:待机后,检测各应用程序的定时器唤醒状态。

[0039] 具体的,在本实施例中,定时器唤醒状态即待机后各应用程序在预设时间内定时唤醒系统的唤醒次数。

[0040] 步骤104:根据检测到的各应用程序的定时器唤醒状态判断应用程序定时器唤醒是否存在异常。

[0041] 真正对齐唤醒的目的是为了达到省电效果,而终端耗电过多主要是由于某些应用频繁设置多次定时器唤醒使得待机时终端被唤醒的次数过多所致。因此,对检测到的各应用程序在预定的时间内定时唤醒系统的次数进行分类,如果应用程序唤醒系统的次数较少,则将应用程序划分为定时器唤醒正常应用程序,执行步骤106;如果该应用程序的唤醒系统的次数过多,则将应用程序划分为定时器唤醒异常应用程序,执行步骤108。

[0042] 步骤106:若定时器唤醒正常,则按定时器唤醒时间唤醒系统,运行应用程序。

[0043] 为了避免出现由于一些在待机后设置较少定时器唤醒次数的应用程序不能正常触发(如,正常应用程序只设置一次定时器唤醒,但是还是被延迟。)遭到投诉的现象,对于定时器唤醒正常的应用程序不做延时处理,应用程序按照定时器唤醒时间正常唤醒系统,通过后台进程推送通知信息,运行应用程序。

[0044] 在本实施例中,通过后台进程推送通知信息。具体的,根据应用程序的属性信息,在系统内建立与其对应的后台服务进程,通过后台服务进程建立与应用程序所属的服务器之间的通信链接。

[0045] 步骤108:若定时器唤醒异常,则将所有定时器唤醒异常应用程序的定时器唤醒时

间设置为待机后固定时间。

[0046] 应用程序频繁唤醒系统会造成耗电过多,对于定时器唤醒异常的应用程序,不响应定时器唤醒对应的操作,并且强制设置定时器唤醒的后续响应的时间为待机后的固定时间,将所有定时器唤醒异常应用程序都延迟到固定时间唤醒系统,做对齐唤醒处理。

[0047] 固定时间的时长可根据实际需要进行具体设置,例如可以为30分钟,1小时,2小时等,综合考虑能够同时兼顾待机省电和通知信息的及时到达即可。另外,还可以结合用户的日常使用情况,按一定的分布情况综合分析得出适合于用户习惯的固定时间。

[0048] 步骤110:在固定时间的时刻点唤醒系统,运行各定时器唤醒异常的应用程序。

[0049] 本实施例中,通过后台进程推送通知信息,运行应用程序。根据应用程序的属性信息,在系统内建立与其对应的后台服务进程,通过后台服务进程建立与应用程序所属的服务器之间的通信链接。

[0050] 具体的,当系统时刻运行到固定时间时刻点时,所有被延时的定时器唤醒异常应用程序在该时刻同时唤醒系统,后台进程推送各定时器唤醒异常应用程序的通知信息,运行定时器唤醒异常的应用程序。当再次运行到下一次固定时间时的刻点时,系统被再次唤醒,此时,后台进程推送在上一次唤醒时刻点到本次唤醒时刻点的时间间隔内各定时器唤醒异常应用程序的通知信息,运行定时器唤醒异常的应用程序。

[0051] 在本实施例中,上述对齐唤醒方法对设置过多定时器唤醒次数的应用做延迟处理,对正常定时器唤醒的应用不做延时处理,实现了对各应用的定时器唤醒次数的针对性管理,既保证了设置正常的应用能够准时唤醒,又实现了对设置频繁唤醒次数的应用程序对齐唤醒,能够很好地节约终端待机耗电,达到省电的效果,且有效保证信息及时送达,实现了后台智能管理。

[0052] 在一个实施例中,在检测各应用程序的定时器唤醒状态步骤中,定时器唤醒状态包括在预设时间内定时器唤醒系统的唤醒次数。

[0053] 具体的,上述的预设时间设置为10分钟~60分钟,在一个较佳实施例中,预设时间为30分钟。预设时间的长短需要从待机耗电量和定时器唤醒状态检测准确性两方面考虑,如果预设时间过短,会导致检测不全面,一方面可能在较短的时间内检测到各应用程序的定时器唤醒次数都较少,干扰后续定时器唤醒异常判断;另一方面,还有可能出现某些程序的定时器还没来得及唤醒而检测就结束了,造成检测遗漏,影响定时器唤醒状态的准确性。因此,需要设置时长合适的预设时间,本实施例中的预设时间时长既能够满足节省待机耗电量的要求,也能够有效保证定时器唤醒状态检测的准确性。

[0054] 如图2所示,在一个实施例中,根据检测到的各应用程序的定时器唤醒状态判断定时器唤醒是否存在异常的步骤包括以下步骤:

[0055] 104a:判断应用程序的类型为定时唤醒程序或非定时唤醒程序。

[0056] 具体的,应用程序包括定时唤醒程序和非定制唤醒程序两种类型,定时唤醒的应用程序会按照定时器设定的时间定时唤醒系统,非定时唤醒的应用程序唤醒系统的时间不是固定的,很多时候可能会即时性的唤醒系统(如微信),针对不同唤醒类型的应用程序,需要设置不同的异常判断条件,以保证定时器唤醒的及时性。

[0057] 104b:若应用程序为定时唤醒程序,则将检测到的预设时间内定时器唤醒系统的唤醒次数与预先设定的第一预设定时器唤醒次数进行比较,若预设时间内定时器的唤醒次

数大于或等于第一预设定时器唤醒次数,则该应用程序为定时器唤醒异常应用程序;否则,若预设时间内定时器的唤醒次数小于第一预设定时器唤醒次数,该应用程序为定时器唤醒正常应用程序。

[0058] 具体的,上述的第一预设定时器唤醒次数为3次~5次,在一个较佳实施例中,第一预设定时器唤醒次数为4次。如,在一个施例中,如果检测到某定时唤醒应用程序在30分钟内唤醒系统达到4次,则对该应用程序的定时器唤醒时间做延迟处理,调整定时器唤醒时间为固定时间。

[0059] 104c:若应用程序为非定时唤醒程序,则将检测到的预设时间内定时器唤醒系统的唤醒次数与预先设定的第二预设定时器唤醒次数进行比较,若预设时间内定时器的唤醒次数大于或等于第二预设定时器唤醒次数,则该应用程序为定时器唤醒异常应用程序;否则,若预设时间内定时器的唤醒次数小于第二预设定时器唤醒次数,该应用程序为定时器唤醒正常应用程序。

[0060] 具体的,上述的第二预设定时器唤醒次数为6次~9次,在一个较佳实施例中,第二预设定时器唤醒次数为7次。如,在一个施例中,如果检测到某非定时唤醒应用程序在30分钟内唤醒系统达到7次,则对该应用程序的定时器唤醒时间做延迟处理,调整定时器唤醒时间为固定时间。

[0061] 在本实施例中,对于定时唤醒程序和非定时唤醒程序设置了不同的异常判断条件,由于非定时唤醒程序的唤醒时间不固定,并且,大多数非定时唤醒程序的通知信息多为即时性信息,为保证信息送达的及时性,设置第二预设定时器唤醒次数大于第一预设定时器唤醒次数。同时,本实施例在每次待机后均进行定时器唤醒异常的判断,能够有效保证准确检测非定时唤醒程序在每次待机时的定时器唤醒次数,从而能够针对每次待机的实时情况做出最适合的对齐唤醒方案。

[0062] 在一个实施例中,在上述的在固定时间的时刻点唤醒系统,运行各定时器唤醒异常的应用程序的步骤之前还包括以下步骤:在上一次唤醒结束至下一次唤醒前的时间间隔内,通过后台进程接管各定时器唤醒异常应用程序的通知信息。

[0063] 例如,固定时间设置为30分钟的固定时长,则在30分钟的间隔内,所有所接收到定时器唤醒异常应用程序的通知信息都交由后台进程接管。具体的,系统根据应用程序的通知信息需求确定资源需求,为与应用程序对应的后台服务进程分配固定的内存空间,根据所述内存空间预设相应的缓存空间存储通知信息。在一个实施例中,后台服务进程还进行内存空间的整理和清空操作,当应用程序在固定时间的时刻点唤醒系统,通过后台进程推送通知信息后,后台进程整理内存空间,并清空缓存空间。

[0064] 在一个实施例中,在根据检测到的各应用程序的定时器唤醒状态判断应用程序定时器唤醒是否存在异常的步骤中还包括以下步骤:若检测某应用程序的定时器唤醒状态失败,则该应用程序为定时器唤醒异常应用程序。

[0065] 不同的应用程序,所采用的编译方式以及编译规则各不相同,且应用程序运行过程中也有可能出现运行异常情况,因此,可能会出现无法采集某个应用程序的唤醒时间的情况,本实施例中,若检测某应用程序的定时器唤醒状态失败,则将该应用程序划分为定时器唤醒异常应用程序,将该应用程序的定时器唤醒时间设置为固定时间。

[0066] 在一个实施例中,在上述的步骤108和步骤110之间还包括以下步骤:在未达到固



定时间时刻点时用户结束待机,所有被延迟的应用程序结束延迟唤醒,发送通知信息。

[0067] 如果用户在没有到达固定时间时刻点时提前结束待机,那么所有被延迟至固定时间时刻点唤醒的应用程序均在待机结束时刻发送通知信息,运行应用程序,以保证信息及送达。

[0068] 如图3所示,其为一个实施例中对齐唤醒系统的结构原理图。

[0069] 一种对齐唤醒装置,包括:

[0070] 唤醒状态检测模块302,用以检测各应用程序的定时器唤醒状态。

[0071] 唤醒异常判断模块304,用以根据检测到的各应用程序的定时器唤醒状态判断应用程序定时器唤醒是否存在异常。

[0072] 对齐唤醒模块306,用以将所有定时器唤醒异常应用程序的定时器唤醒时间设置为待机后固定时间。

[0073] 唤醒触发模块308,用以唤醒系统,运行应用程序。

[0074] 在一个实施例中,唤醒状态检测模块302用以检测预设时间内定时器唤醒系统的唤醒次数。

[0075] 在一个实施例中,唤醒异常判断模块304包括:应用程序类型判断子模块,用以判断应用程序的类型为定时唤醒程序或非定时唤醒程序。存储子模块,用以存储预设时间、第一预设定时器唤醒次数和第二预设定时器唤醒次数。第一唤醒异常判断子模块,用以判断定时唤醒程序的定时器唤醒是否存在异常,若应用程序为定时唤醒程序,所述第一唤醒异常判断子模块将检测到的预设时间内定时器唤醒系统的唤醒次数与的第一预设定时器唤醒次数进行比较,若预设时间内定时器的唤醒次数大于或等于所述第一预设定时器唤醒次数,则该应用程序为定时器唤醒异常应用程序;若预设时间内定时器的唤醒次数小于第一预设定时器唤醒次数,该应用程序为定时器唤醒正常应用程序。第二唤醒异常判断子模块,用以判断非定时唤醒程序的定时器唤醒是否存在异常,若应用程序为非定时唤醒程序,所述第二唤醒异常判断子模块将检测到的预设时间内定时器唤醒系统的唤醒次数与第二预设定时器唤醒次数进行比较,若预设时间内定时器的唤醒次数大于或等于所述第二预设定时器唤醒次数,则所述应用程序为定时器唤醒异常应用程序;若预设时间内定时器的唤醒次数小于第二预设定时器唤醒次数,所述应用程序为定时器唤醒正常应用程序。

[0076] 在一个实施例中,唤醒异常判断模块304还包括唤醒状态统计子模块,唤醒状态统计子模块用以检测并统计预设时间内定时器唤醒系统的唤醒状态,若检测应用程序的定时器唤醒状态成功,则所述唤醒状态检测子模块统计预设时间内定时器唤醒系统的唤醒次数;若检测某应用程序的定时器唤醒状态失败,则所述唤醒状态检测子模块定义该应用程序为定时器唤醒异常应用程序。

[0077] 以下结合附图列举优选实施例对上述的唤醒异常判断模块304进行具体说明。如图4所示,在一个实施例中,上述的唤醒异常判断模块304包括:

[0078] 应用程序类型判断子模块304a,用以判断应用程序的类型为定时唤醒程序或非定时唤醒程序。

[0079] 存储子模块304b,用以存储预设时间、第一预设定时器唤醒次数和第二预设定时器唤醒次数。

[0080] 具体的,预设时间设置为10分钟~60分钟,在一个较佳实施例中,预设时间为30分

钟。第一预设定时器唤醒次数为3次~5次,在一个较佳实施例中,第一预设定时器唤醒次数为4次。第二预设定时器唤醒次数为6次~9次,在一个较佳实施例中,第二预设定时器唤醒次数为7次。

[0081] 唤醒状态统计子模块304c,用以检测并统计预设时间内定时器唤醒系统的唤醒状态。

[0082] 具体的,唤醒状态检测子模块304c检测并统计30分钟内定时器唤醒系统的唤醒次数。若检测应用程序的定时器唤醒状态成功,则统计30分钟内定时器唤醒系统的唤醒次数;若检测某应用程序的定时器唤醒状态失败,则定义该应用程序为定时器唤醒异常应用程序。

[0083] 第一唤醒异常判断子模块304d,用以判断定时唤醒程序的定时器唤醒是否存在异常。

[0084] 具体的,第一唤醒异常判断子模块304d将唤醒状态检测子模块304c检测到的30分钟内定时唤醒程序的定时器唤醒系统的唤醒次数与预先设定的第一预设定时器唤醒次数进行比较,若30分钟内定时器的唤醒次数大于或等于4次,则该应用程序为定时器唤醒异常应用程序;若30分钟内定时器的唤醒次数小于4次,该应用程序为定时器唤醒正常应用程序。

[0085] 第二唤醒异常判断子模块304e,用以判断非定时唤醒程序的定时器唤醒是否存在异常。

[0086] 具体的,第二唤醒异常判断子模块304e将唤醒状态检测子模块304c检测到的30分钟内非定时唤醒程序的定时器唤醒系统的唤醒次数与预先设定的第二预设定时器唤醒次数进行比较,若30分钟内定时器的唤醒次数大于或等于7次,则应用程序为定时器唤醒异常应用程序;若30分钟内定时器的唤醒次数小于7次,应用程序为定时器唤醒正常应用程序。

[0087] 在一个实施例中,上述的对齐唤醒装置还包括:通知信息管理模块,用以在上一次唤醒结束至下一次唤醒前的时间间隔内,通过后台进程接管各定时器唤醒异常应用程序的通知信息。

[0088] 以上所述实施例的各技术特征可以进行任意的组合,为使描述简洁,未对上述实施例中的各个技术特征所有可能的组合都进行描述,然而,只要这些技术特征的组合不存在矛盾,都应当认为是本说明书记载的范围。

[0089] 以上所述实施例仅表达了本发明的几种实施方式,其描述较为具体和详细,但并不能因此而理解为对发明专利范围的限制。应当指出的是,对于本领域的普通技术人员来说,在不脱离本发明构思的前提下,还可以做出若干变形和改进,这些都属于本发明的保护范围。因此,本发明专利的保护范围应以所附权利要求为准。

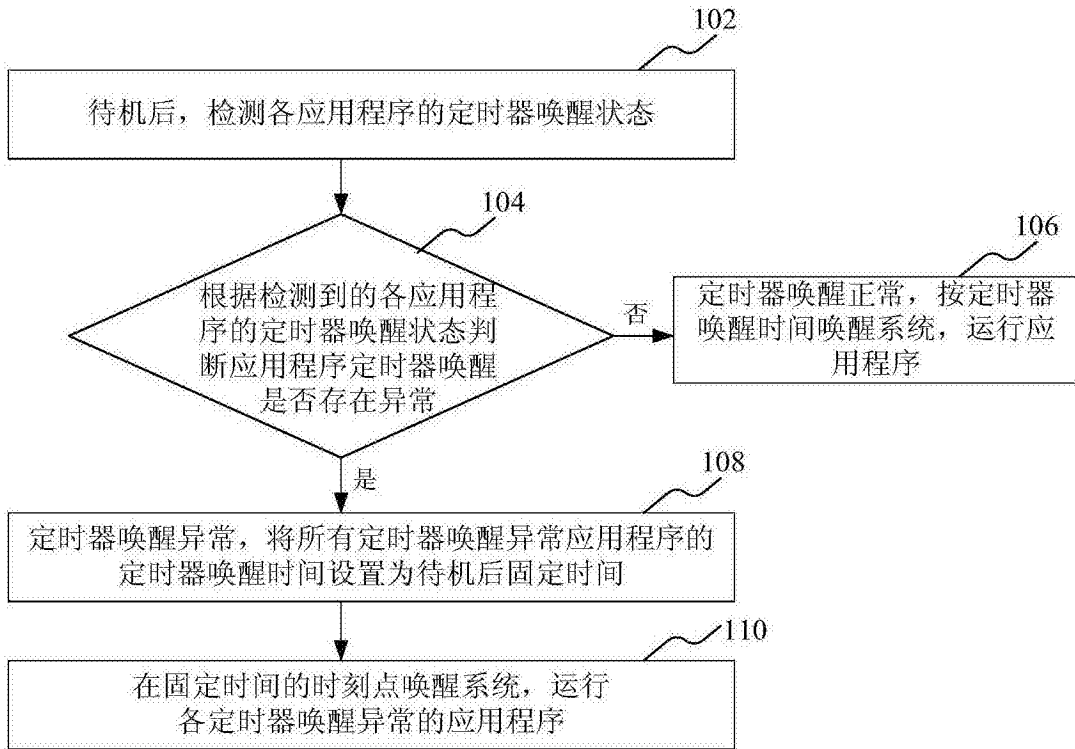


图1

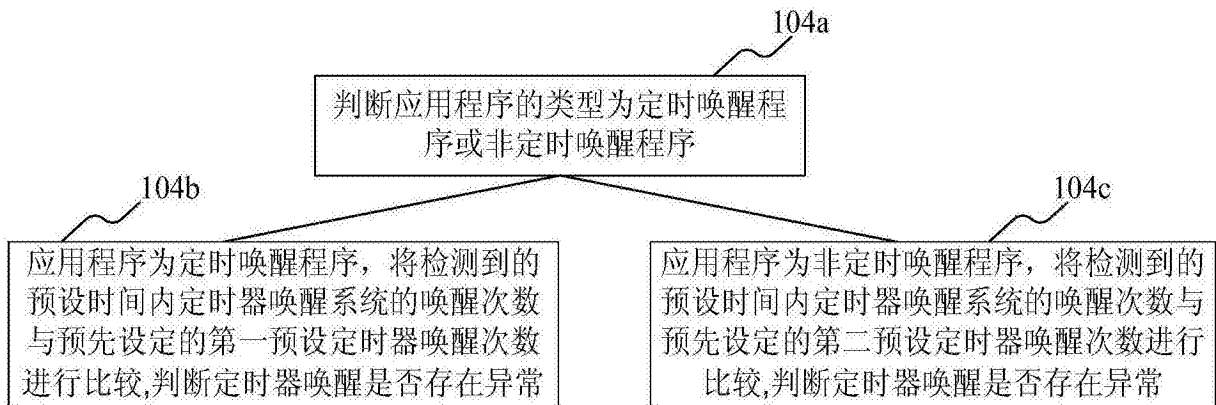


图2

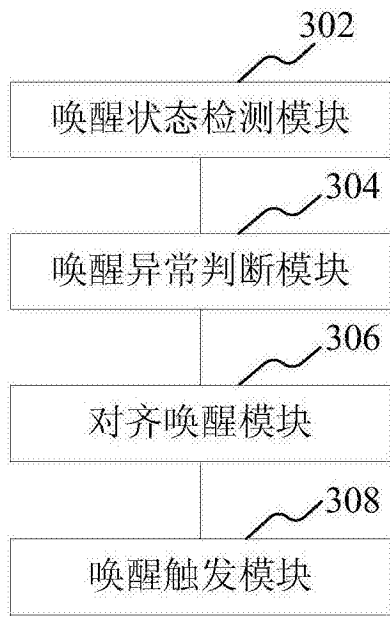


图3

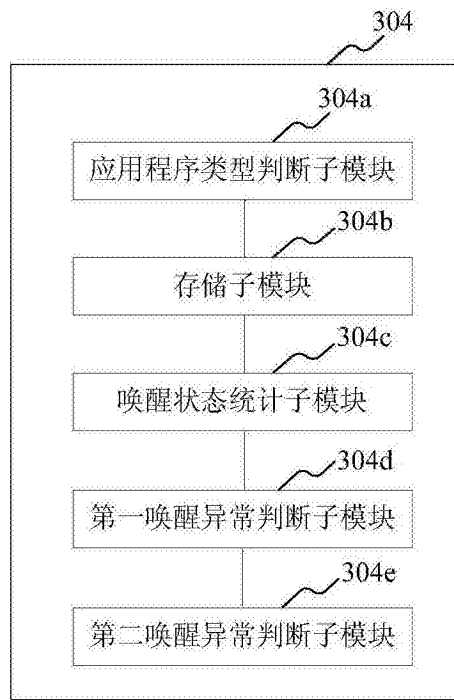


图4