



(12) 发明专利

(10) 授权公告号 CN 110989816 B

(45) 授权公告日 2022.01.14

(21) 申请号 201910907466.2

G06F 1/329 (2019.01)

(22) 申请日 2019.09.24

G06F 3/16 (2006.01)

(65) 同一申请的已公布的文献号  
申请公布号 CN 110989816 A

H04W 52/02 (2009.01)

H04M 1/60 (2006.01)

(43) 申请公布日 2020.04.10

(56) 对比文件

(73) 专利权人 华为技术有限公司  
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

CN 106937162 A, 2017.07.07

CN 106937162 A, 2017.07.07

CN 106887244 A, 2017.06.23

CN 1937403 A, 2007.03.28

(72) 发明人 陈亮 陈谭坤 张威

CN 106200872 A, 2016.12.07

CN 102339619 A, 2012.02.01

(74) 专利代理机构 北京同立钧成知识产权代理有限公司 11205

CN 109673042 A, 2019.04.23

CN 107360386 A, 2017.11.17

代理人 荣甜甜 刘芳

US 2017237986 A1, 2017.08.17

审查员 唐丹颖

(51) Int. Cl.

G06F 1/3206 (2019.01)

G06F 1/3234 (2019.01)

权利要求书1页 说明书14页 附图6页

(54) 发明名称

应用的处理方法、装置以及存储介质

(57) 摘要

本申请实施例提供一种应用的处理方法、装置以及存储介质,该方法包括:检测应用对应的音轨AudioTrack的音量设置接口,获取应用的设置音量;若应用的设置音量小于第一阈值且应用的系统设置音量大于第二阈值,则确定应用处于无效放音状态;对应用的放音行为进行管控。本申请实施例通过应用的设置音量来确定应用的放音状态,实现了对应用的无效放音的监控,当识别出应用处于无效放音状态时可以停止应用的无效放音,从而减少功耗,提高用户性能体验。



1. 一种应用的处理方法,其特征在于,包括:

检测应用对应的音轨AudioTrack的音量设置接口,获取所述应用的设置音量;其中,所述应用为后台应用,所述应用的设置音量为音轨音量;

若所述应用的设置音量小于第一阈值且所述应用所在终端的当前系统设置音量大于第二阈值,则确定所述应用处于无效放音状态;所述应用处于无效放音状态表示所述应用正在进行恶意放音行为;

对所述应用的放音行为进行管控。

2. 根据权利要求1所述的方法,其特征在于,还包括:

检测应用对应的AudioFlinger的设置最终音量接口,获取所述应用的实际播放音量;

若所述应用的实际播放音量小于第三阈值且所述应用的系统实际音量大于第四阈值,则确定所述应用处于无效放音状态。

3. 根据权利要求1所述的方法,其特征在于,还包括:

获取所述AudioTrack连续调用write接口输入静音音频帧的数量;

若所述静音音频帧的数量大于第五阈值,则确定所述应用处于无效放音状态。

4. 根据权利要求1所述的方法,其特征在于,所述对所述应用的放音行为进行管控,包括:

关闭所述应用的放音进程。

5. 根据权利要求4所述的方法,其特征在于,在所述关闭所述应用的放音进程之后,还包括:

若再次检测到所述应用处于无效放音状态,则关闭所述应用的前台界面并停止所述应用的后台服务。

6. 根据权利要求5所述的方法,其特征在于,在所述关闭所述应用的前台界面并停止所述应用的后台服务之后,还包括:

若再次检测到所述应用处于无效放音状态,则强制停用所述应用。

7. 根据权利要求1-6任一项所述的方法,其特征在于,在所述检测应用对应的音轨AudioTrack的音量设置接口,获取所述应用的设置音量之前,还包括:

若检测到所述AudioTrack的对象被创建,则启动所述应用的无效放音检测。

8. 根据权利要求1-6任一项所述的方法,其特征在于,还包括:

若检测到所述AudioTrack的对象被移除或销毁,则停止所述应用的无效放音检测。

9. 一种应用的处理装置,其特征在于,所述应用的处理装置包括:处理器和存储器;

其中,所述存储器用于存储程序;

所述处理器用于执行权利要求1-7任一项所述的应用的处理方法。

10. 一种存储介质,其上存储有计算机程序,其特征在于,包括:该程序被处理器执行时实现权利要求1-7任一项所述的方法。

## 应用的处理方法、装置以及存储介质

### 技术领域

[0001] 本申请实施例涉及通信领域,尤其涉及一种应用的处理方法、装置以及存储介质。

### 背景技术

[0002] 应用放音指电子设备上安装的应用通过电子设备运行的操作系统的系统服务进行声音播放的行为。操作系统以内置的媒体播放器(MediaPlayer)和音轨(AudioTrack),为应用提供声音播放功能。MediaPlayer将声音文件解码后的脉宽调制(pulse code modulation,PCM)音频数据流传递给AudioTrack,由AudioTrack进行播放。然而,电子设备的中央处理器(central processing unit,CPU)、内存、电池等资源都是有限的,当操作系统需要消耗大量资源时,有可能造成资源不足的情况。因此,为了保证用户操控体验,操作系统需要对后台应用的放音行为进行管控,将放音的应用冻结或清理。

[0003] 现有技术中,通常是通过监控播放器(play back activity monitor)对应用播放音频进行监控。监控播放器里保存了公开在软件开发包(software development kit, SDK)里的所有播放器的实例。因此,可以通过查询接口查询正在播放音频的应用,也可以通过打点上报方式进行监控。

[0004] 然而,现有的监控播放器只能监控应用是否进行放音行为,而无法监控播放静音文件和低音量播放的无效放音行为。

### 发明内容

[0005] 本申请实施例提供一种应用的处理方法、装置以及存储介质,用以解决现有技术中无法监控播放静音文件和低音量播放的无效放音行为的问题。

[0006] 第一方面,本申请实施例提供一种应用的处理方法,该方法可以应用于终端、也可以应用于终端中的芯片。下面以应用于终端为例对该方法进行描述,该方法中,检测应用对应的音轨AudioTrack的音量设置接口,获取所述应用的设置音量;若所述应用的设置音量小于第一阈值且所述应用的系统设置音量大于第二阈值,则确定所述应用处于无效放音状态;对所述应用的放音行为进行管控。

[0007] 通过第一方面提供的应用的处理方法,可以根据应用输出的设置音量,确定应用处于无效放音状态,若是应用处于无效放音状态,则停止应用的无效放音。该方法中通过获取应用的设置音量来确定应用的放音状态,实现了对应用的无效放音的监控,当识别出应用处于无效放音状态时可以停止应用的无效放音,从而减少功耗,提高用户性能体验。

[0008] 在一种可实施的方式中,应用的处理方法还包括:检测应用对应的AudioFlinger的设置最终音量接口,获取所述应用的实际播放音量;若所述应用的实际播放音量小于第三阈值且所述应用的系统实际音量大于第四阈值,则确定所述应用处于无效放音状态。

[0009] 通过该可实施的方式提供的应用的处理方法,可以通过应用的实际播放音量来确定应用是否处于无效放音状态,实现了对应用的无效放音的监控,从而减少功耗,提高用户性能体验。

[0010] 在一种可实施的方式中,应用的处理方法还包括:获取所述AudioTrack连续调用wirte 接口输入静音音频帧的数量;若所述静音音频帧的数量大于第五阈值,则确定所述应用处于 无效放音状态。

[0011] 通过该可实施的方式提供的应用的处理方法,可以通过连续调用wirte接口输入静音音频 帧的数量确定应用是否处于无效放音状态,实现了对应用的无效放音的监控,从而减少功 耗,提高用户性能体验。

[0012] 在一种可实施的方式中,所述对所述应用的放音行为进行管控,包括:关闭所述应 用的 放音进程。

[0013] 通过该可实施的方式提供的应用的处理方法,可以通过关闭所述应用的放音进程 来停止 无效放音应用的放音行为,从而减少功耗,提高用户性能体验。

[0014] 在一种可实施的方式中,所在所述关闭所述应用的放音进程之后,还包括:若再次 检测 到所述应用处于无效放音状态,则关闭所述应用的前台界面并停止所述应用的后台 服务。

[0015] 通过该可实施的方式提供的应用的处理方法,若关闭应用的放音进程后,应用再 次进行 无效放音,可以通过关闭应用的前台界面并停止应用的后台服务来对应用进行更 严格的管控, 从而减少功耗,提高用户性能体验。

[0016] 在一种可实施的方式中,在所述关闭所述应用的前台界面并停止所述应用的后台 服务之 后,还包括:若再次检测到所述应用处于无效放音状态,则强制停用所述应用。

[0017] 通过该可实施的方式提供的应用的处理方法,若关闭应用的前台界面并停止应 用的后 台 服务后,应用再次进行无效放音,则可以确定应用为恶意放音应用,可以强制停用 所述应用, 从而减少功耗,提高用户性能体验。

[0018] 在一种可实施的方式中,在所述检测应用对应的音轨AudioTrack的音量设置接 口,获取 所述应用的设置音量之前,还包括:若检测到所述AudioTrack的对象被创建,则启 动所述应 用的无效放音检测。

[0019] 通过该可实施的方式提供的应用的处理方法,可以根据AudioTrack的对象的创 建,确定 应用开始进行放音,进而开始对应用进行无效放音检测。

[0020] 在一种可实施的方式中,应用的处理方法还包括:若检测到所述AudioTrack的对 象被 移除或销毁,则停止所述应用的无效放音检测。

[0021] 通过该可实施的方式提供的应用的处理方法,可以根据AudioTrack的对象被移 除或 销毁, 确定应用结束放音,进而结束对应用进行无效放音检测。

[0022] 第二方面,本申请实施例提供一种应用处理的终端设备的处理装置,所述终端 设备包 括: 处理器和存储器,其中,所述存储器用于存储程序;所述处理器用于执行第一个方 面所述 的 任一项应用的处理方法。

[0023] 第三方面,本申请实施例提供一种存储介质,其上存储有计算机程序,包括:该 程序 被处理器执行时上述第一方面或第一方面的各种实施方式的应用的处理方法。

[0024] 本申请实施例提供的应用的处理方法、装置以及存储介质,通过检测应用对应 的音 轨AudioTrack的音量设置接口,获取应用的设置音量;若应用的设置音量小于第一阈 值且 应用的系统设置音量大于第二阈值,则确定应用处于无效放音状态;对应用的放音行 为进 行管控。该方法中通过应用的设置音量来确定应用的放音状态,实现了对应用的无效放

音的监控,当识别出应用处于无效放音状态时可以停止应用的无效放音,从而减少功耗,提高用户性能体验。

### 附图说明

- [0025] 图1为本申请实施例提供的一种应用的处理方法的应用场景的示意图;
- [0026] 图2为本申请实施例提供的一种终端设备的放音示意图;
- [0027] 图3为本申请实施例提供的应用的处理方法的实施例一的流程示意图;
- [0028] 图4为本申请实施例提供的一种终端设备的界面示意图;
- [0029] 图5为本申请实施例提供的应用的处理方法的实施例二的流程示意图;
- [0030] 图6为本申请实施例提供的应用的处理方法的实施例三的流程示意图;
- [0031] 图7为本申请实施例提供的应用的处理方法的实施例四的流程示意图;
- [0032] 图8为本申请实施例提供的一种应用的管控装置示意图;
- [0033] 图9为本申请实施例提供的应用的处理方法的实施例五的流程示意图;
- [0034] 图10为本申请实施例提供的一种应用处理的终端设备的处理装置;
- [0035] 图11为本申请实施例提供的一种电子设备的结构示意图;
- [0036] 图12为本申请实施例提供的应用的处理装置为手机时的结构框图。

### 具体实施方式

[0037] 应当理解,本申请实施例中的术语“第一”、“第二”等是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。应该理解这样使用的术语在适当情况下可以互换,以便本申请实施例能够以除图示或描述的那些以外的顺序实施。

[0038] 应用程序是移动终端多功能化的重要组成部分,为用户提供了各种个性化的功能。应用放音指应用程序通过系统服务进行声音播放的状态。例如:用户在选择音乐和听书类应用进行音乐播放和内容收听时,可以将这些应用切换到后台同时进行其他活动,例如:聊天、游戏、文字处理等。在上述情况下,发出声音的应用处于一种用户期望地正常放音状态。

[0039] 操作系统以内置的媒体播放器(MediaPlayer)和音轨(AudioTrack),为应用提供声音播放功能。MediaPlayer将声音文件解码后的脉宽调制(pulse code modulation, PCM)音频数据流传递给AudioTrack,由AudioTrack进行播放。

[0040] 然而,智能终端的CPU、内存、电池等资源都是有限的,当系统需要消耗大量资源时,有可能造成系统资源不足,因此,为了保证用户操控体验,操作系统需要对后台应用的放音行为进行管控,在资源不足时,将放音的应用冻结或清理。应用管控常见的两种方式是冻结和清理,应用冻结是指将应用限制起来不允许其活动,使其不影响系统功耗及性能。应用清理是指将应用强制停止。

[0041] 现有技术中,通常是通过监控播放器(play back activity monitor)对应用播放音频进行监控。监控播放器里保存了公开在软件开发包(software development kit, SDK)里的所有播放器的实例。因此,可以通过查询接口查询正在播放音频的应用,也可以通过打点上报方式进行监控。

[0042] 然而,现有的监控播放器只能监控应用是否进行放音行为,而无法监控播放静音

文件和低音量播放的无效放音行为。应用恶意进行无效放音，往往会增加系统的能耗，降低用户的性能体验。

[0043] 考虑到上述问题，本申请实施例提供了一种应用的处理方法，通过检测应用的放音事件，获取应用输出的音频帧或应用对应的音量，并根据应用输出的音频帧或应用对应的音量，确定应用处于无效放音状态，随之停止应用的放音事件，从而实现对应用的无效放音的监控，当识别出应用处于无效放音状态时可以停止应用的无效放音，从而减少功耗，提高用户性能体验。

[0044] 图1为本申请实施例提供的一种应用的处理方法的应用场景示意图。图2为本申请实施例提供的一种终端设备的放音示意图。如图1所示，应用的处理系统可以包括：终端设备101和应用服务器102。其中，应用服务器102可以为一个也可以为多个，终端设备101中的应用可以从对应的应用服务器102中获取音频资源，并通过终端设备101中的音频器件进行播放。

[0045] 如图2所示，终端设备101在进行放音时，终端设备101通过放音的应用创建一个AudioTrack类的实例，该AudioTrack可以处理上述应用输出的音频数据流。当AudioTrack创建时，音频护圈(AudioFlinger)会有接收到AudioTrack创建的通知，同时，AudioFlinger对上层接口接收到的AudioTrack进行混音，并通过AudioFlinger的下层接口控制终端设备101的音频器件进行放音。通常，AudioTrack和AudioFlinger并不在同一个进程中，它们通过安卓中的捆绑(binder)机制建立联系。

[0046] 其中，AudioTrack和AudioFlinger均包含有一些放音特征。示例性的，AudioTrack可以包含有AudioTrack的创建与销毁、静音音频帧的数量、应用设置音量，AudioFlinger可以包含系统实际音量。

[0047] 终端设备101的操作系统以内置系统服务的方式为应用提供声音播放功能，内置系统服务中包括媒体播放器(MediaPlayer)和AudioTrack两种方式，均可以为应用提供声音播放功能。其中，AudioTrack没有解码功能，只能播放解码之后的音频数据流。MediaPlayer可以将声音文件解码，并将解码后的音频数据流传递给AudioTrack处理，进而让AudioTrack播放解码之后的音频数据流。因此，应用在进行放音时，均会输出音频数据流至AudioTrack。

[0048] 在Android Framework的音频子系统中，每一个应用输出的音频数据流对应着一个AudioTrack类的一个实例。每个AudioTrack在创建时会通知AudioFlinger，AudioFlinger为每个AudioTrack类的实例建立与一个具体的工作线程的对应关系，并通知这个工作线程创建了一个Track对象与这个AudioTrack进行对应。AudioFlinger把所有的AudioTrack进行混音，然后输送到终端设备101的音频器件进行播放。

[0049] 本申请实施例中，可以对应用的放音进行监控，当应用处于无效放音状态时，可以停止应用的无效放音，从而减少功耗，提高用户性能体验。

[0050] 本申请实施例可以应用于具有放音功能的终端设备中，终端设备可包括手机、电脑、智能手表等，本申请对此不做限制。

[0051] 示例性的，以终端设备为手机为例，当系统前台进行视频播放或进行游戏时，若系统CPU内存资源供给不足，为了释放该应用占用的内存资源，优先保障用户感知的前台应用体验，可以检测应用的放音状态，优先清理处于无效放音状态的应用，强制停止应用的

放音事件。当系统灭屏待机时,若应用进行无效放音,可能导致系统无法休眠不断消耗电量。因此,可以关闭应用的放音事件,释放音频资源,让系统顺利休眠以节约电量。

[0052] 可以理解,本申请实施例提供的方法,可以适用于任一终端设备对于应用的放音进行监控的场景。本申请实施例中应用的处理方法的执行主体为应用的处理装置,该应用的处理装置可以由任意的软件和/或硬件实现,可以是终端设备的部分或全部,例如可以是终端设备中的处理器。

[0053] 下面以集成或安装有相关执行代码的处理器为例,以具体地实施例对本申请实施例的技术方案进行详细说明。下面这几个具体的实施例可以相互结合,对于相同或相似的概念或过程可能在某些实施例不再赘述。

[0054] 图3为本申请实施例提供的应用的处理方法的实施例一的流程示意图。如图3所示,该应用的处理方法包括:

[0055] S201、检测应用对应的音轨AudioTrack的音量设置接口,获取应用的设置音量。

[0056] 本实施例中,当应用开始进行放音时,处理器可以开始对应用的放音事件进行检测,以便在终端设备的系统资源不足时,停止应用的放音事件。由于无效放音行为为播放静音文件或低音量播放的行为,其往往为恶意放音行为,会造成系统资源被占用。因此,对于应用放音进行监控时,可以进一步确定应用是否处于无效放音状态,以便优先停止处于无效放音状态的应用的放音事件。

[0057] 在正常的放音时,应用播放的音量为人耳可识别的音量。当音量低于人耳能正常听到的音量时,可以确定该应用正在进行无效放音。因此,可以通过检测应用设置音量来确定应用是否处于无效放音状态。因此,可将应用设置音量作为放音特征。

[0058] 在一些实施例中,应用播放声音时,会调用AudioTrack的音量设置(setVolume)接口,进行该应用放音行为的音量设置,此时应用可以从应用对应的AudioFlinger的设置最终音量接口,获取应用的实际播放音量。

[0059] S202、若应用的设置音量小于第一阈值且应用的系统设置音量大于第二阈值,则确定应用处于无效放音状态。

[0060] 本步骤中,可以将应用设置音量和第一阈值进行比较,若应用设置音量小于第二阈值,且系统设置音量大于第二阈值,则确定应用处于无效放音状态。

[0061] 在一些实施例中,系统检测到AudioTrack的setVolume接口中设置音量小于预先设定的第一阈值,且当前系统设置音量大于预先设定的第二阈值时,当应用的放音事件持续预定的时间后,可以判定关联到此AudioTrack的应用在进行无效播放。

[0062] 示例性的,在通过应用设置音量确定应用处于无效放音状态前,首先需要获取系统设置音量,并确定当前的系统设置音量未设置成静音或设置的过低。具体的,可以将系统音量和预设的第二阈值进行比较,若系统音量大于第二阈值,则表明当前的系统设置音量未设置成静音或设置的过低。在确定系统设置音量大于第二阈值后,此时可以进一步确定应用设置音量是否为人耳可识别的音量。具体的,可以将应用设置音量和第一阈值进行比较,若应用设置音量小于第一阈值,则表明应用设置的播放音量为人耳不可识别的音量,进而可以确定应用处于无效放音状态。若应用设置音量不小于第一阈值,则表明应用处于正常放音状态。

[0063] 其中,应用设置音量是由AudioTrack的setVolume接口所设置的,其可以是一种音

轨 (track) 音量,是一个取值为0~1.0的浮点数。相应的,上述第一阈值的取值也相应的为0~1.0 的浮点数。需要说明的是,本申请实施例对于第一阈值的设置不做限制,取0~1.0 之间的值 即可。同样的,本申请实施例对于第二阈值的设置也不做限制。

[0064] 上述的系统设置音量可以由用户进行手动设置。本申请实施例对于如何设置系统设置音 量不做限制,在一种可实施方式中,可以通过终端设备上的调音按钮逐步增加或者减少系统 设置音量。在另一种可实施方式中,可以在终端上直接输入具体的音量值进行设置。

[0065] S203、对应用的放音行为进行管控。

[0066] 本步骤中,由于无效放音会占用系统内存,增加功耗,降低用户的性能体验。因此,当 检测到应用处于无效放音时,处理器可以停止应用的放音事件。

[0067] 其中,停止应用的放音事件的管控方式通常采用冻结应用和/或清理应用的方式。冻结应 用是将应用限制起来不允许其活动,可以包括:关闭应用的放音进程和关闭应用的前台界面 并停止应用的后台服务,清理应用是强制停用应用。

[0068] 需要说明的是,本申请实施例对于如何停止应用的无效放音不做限制,可以采用 一种或 多种管控方式停止应用的无效放音。

[0069] 在一些实施例中,可以采用分级处理的方式停止应用的无效放音。示例性的,为了 能够 最大程度不影响应用功能和减少系统消耗,可以采用提出了对应用的三级管控方式。其中, 第一级管控可以为关闭 (kill) 应用的放音进程;第二级管控可以为关闭应用的前台界面 (RemoveTask) 并停止应用的后台服务 (StopService);第三级管控可以为强制停用应用 (ForceStop)。

[0070] 在一些实施例中,处理器可以根据终端设备的运行状态停止应用的放音事件。

[0071] 示例性的,当系统前台进行视频播放或进行游戏时,系统CPU内存资源供给不足时。应 当把无妨放音的应用列为优先清理的对象,释放其占用的内存资源,优先保障用户感知的前 台应用体验。

[0072] 示例性的,当系统灭屏待机时,应用进行无效放音,导致系统无法休眠不断消耗电 量。应当把无效放音的应用进行管控,释放音频资源,让系统顺利休眠以节约电量。

[0073] 图4为本申请实施例提供的一种终端设备的界面示意图。如图4所示,在一种可选的实 施方式中,当处理器确定应用进行无效放音时,还可以在电子设备的用户界面上弹出提醒框, 以提醒用户并通过与用户交互来接收用户的指示以确定是否停止该应用的放音事件。

[0074] 本申请实施例提供的应用的处理方法,通过检测应用对应的音轨AudioTrack的音 量 设置接口,获取应用的设置音量;若应用的设置音量小于第一阈值且应用的系统设置音 量大于第二阈值,则确定应用处于无效放音状态;对应用的放音行为进行管控。该方法 中通过应用的设置音量来确定应用的放音状态,实现了对应用的无效放音的监控,当识别出 应用处于无效放音状态时可以停止应用的无效放音,从而减少功耗,提高用户性能体验。

[0075] 在上述实施例的基础上,除了可以通过应用的设置音量,还可以通过实际播放音 量确定 应用处于无效放音状态。图5为本申请实施例提供的应用的处理方法的实施例二的 流程示 意图。如图5所示,该应用的处理方法,还包括:

[0076] S301、检测应用对应的AudioFlinger的设置最终音量接口,获取应用的实际播放

音量。

[0077] 下面对于上述应用设置音量和上述实际播放音量的关系进行进行说明。实际播放音量，可以是终端设备在实际进行放音时的音量。实际播放音量可以结合主设备音量，流音量以及 线程(track) 音量而成。其中，track音量由放音事件对应的应用通过AudioTrack中的设置最 终音量setVolume接口来设置。应用设置音量信息可以是上述track音量。此外，用户还可以 设置系统设置音量，并根据系统设置音量生成系统实际音量。上述系统设置音量可以作为应 用设置音量的上限值，同样的，系统实际音量可以作为实际播放音量的上限值。

[0078] 示例性的，对于放音特征中的实际播放音量信息，处理器可以在AudioFlinger的设置最 终音量(setFinalVolume) 接口中获取实际播放音量信息。

[0079] S302、若应用的实际播放音量小于第三阈值且应用的系统实际音量大于第四阈值，则确 定应用处于无效放音状态。

[0080] 在本步骤中，可以将实际播放音量和第四阈值进行比较，若实际播放音量小第三阈值， 且系统实际音量大于第四阈值，则确定应用处于无效放音状态。

[0081] 在一些实施例中，系统检测到AudioFlinger的setFinalVolume接口中实际播放音量为小 于预先设定的第三阈值，且当前系统实际音量大于预先设定的第四阈值时，当应用的放音事 件持续预定的时间后，可以判定关联到此AudioTrack的应用在进行无效播放。

[0082] 示例性的，在通过实际播放音量确定应用处于无效放音状态前，首先需要获取系 统实际 音量，并确定当前的系统实际音量未设置成静音或设置的过低。具体的，可以将系 统实际音 量和预设的第四阈值进行比较，若系统实际音量大于第四阈值，则表明当前的系 统实际音量 未设置成静音或设置的过低。若系统实际音量大于第四阈值，可以进一步确定 实际播放音量 是否为人耳可识别的音量。具体的，可以将实际播放音量和第三阈值紧进行比较，若实际播 放音量小于第三阈值，则表明实际播放音量为人耳不可识别的音量，进而 可以确定应用处于 无效放音状态。若实际播放音量大于第三阈值，则表明应用为正常放音 状态。

[0083] 其中，实际播放音量可以由AudioFlinger的setFinalVolume接口监测获取，实际 播放音 量可以是结合了主设备音量，流音量以及track音量的最终音量。

[0084] 需要说明的是，本申请实施例对于第三阈值和第四阈值的设置不做限制，可以根 据实际 情况具体设置。同时，由于实际播放音量和应用设置音量的差异，第一阈值不等于 第三阈值， 第二阈值也不等于第四阈值。

[0085] S303、对应用的放音行为进行管控。

[0086] 本实施例中，步骤S303的具体实现过程和实现原理与实施例一中步骤S203的类 似，此 处不再赘述。

[0087] 本申请实施例提供的应用的处理方法，通过该可实施的方式提供的应用的处理方 法， 可以通过应用的实际播放音量来确定应用是否处于无效放音状态，实现了对应用的无 效放音 的监控，从而减少功耗，提高用户性能体验。

[0088] 在上述实施例的基础上，除了可以通过应用的设置音量，还可以通过连续输入静 音音 频帧的数量确定应用处于无效放音状态。图6为本申请实施例提供的应用的处理方法 的实 施例三的流程示意图。如图6所示，该应用的处理方法，还包括：

[0089] S401、获取AudioTrack连续调用write接口输入静音音频帧的数量；

[0090] 下面对于静音音频帧进行说明。在正常的应用放音时，应用输出至AudioTrack的音频数据流中包含有各种音频帧，AudioTrack会连续调用write接口输入音频帧。若音频数据流中包含有少量静音音频帧，属于放音事件中的正常情况。而在无效放音时，应用会播放静音文件，相应的，应用输出至AudioTrack的音频数据流中会包含有大段连续的静音音频帧。基于此，处理器可以获取应用输出的音频帧，并从其中识别出静音音频帧。通过连续输出静音音频帧的数量可以区分正常放音和无效放音判断应用是否在播放静音文件，进而确定应用是否处于无效放音状态。因此，可以将静音音频帧的信息作为一个放音特征。

[0091] 示例性地，对于获取放音特征中的静音音频帧信息，处理器可以在应用创建的AudioTrack类的实例中检测该应用输出的音频数据流，并根据音频数据流获取AudioTrack连续调用write接口写入的静音音频帧信息。

[0092] S402、若静音音频帧的数量大于第五阈值，则确定应用处于无效放音状态。

[0093] 在本步骤中，可以根据静音音频帧确定应用是否进行无效放音，若静音音频帧的数量大于第五阈值，则确定应用处于无效放音状态。

[0094] 示例性的，应用播放音频时，无论采用MediaPlayer还是AudioTrack来处理解码之后的PCM数据流播放音频，均会创建AudioTrack来处理解码之后的PCM数据流。处理器可以在AudioTrack内监测PCM数据流，确定AudioTrack调用write接口连续输出静音音频帧的数量，并将静音音频帧的数量和预设的第五阈值进行比较。若处理器确定的AudioTrack连续调用write接口连续输出静音音频帧的数量大于第五阈值，则表明该应用的放音事件连续输出的静音音频帧超过了正常放音时连续输出静音音频帧的最大数量。此时，可以确定应用正在播放的音频文件为静音文件，应用处在无效放音状态。若处理器确定的应用连续输出至第一线程的静音音频帧的数量小于第五阈值，则表明为正常的应用放音。

[0095] 需要说明的是，本申请实施例对于第五阈值的设置不做限制，可以根据实际情况具体设置。例如：可以设置静音音频帧的数量的阈值为100帧，当应用连续输出的静音音频帧的数量超过100帧时，可以确定应用处于无效放音状态。

[0096] S403、对应用的放音行为进行管控。

[0097] 本实施例中，步骤S403的具体实现过程和实现原理与实施例一中步骤S203的类似，此处不再赘述。

[0098] 本申请实施例提供的应用的处理方法，通过该可实施的方式提供的应用的处理方法，可以通过连续调用write接口输入静音音频帧的数量确定应用是否处于无效放音状态，实现了对应用的无效放音的监控，从而减少功耗，提高用户性能体验。

[0099] 下面结合图7对本申请实施例提供的应用的处理方法中如何开始对应用进行无效放音识别以及如何停止对应用进行无效放音识别进行详细说明。图7为本申请实施例提供的应用的处理方法的实施例四的流程示意图，如图7所示，本实施例提供的应用的处理方法可以包括：

[0100] S501、若检测到AudioTrack的对象被创建，则启动应用的无效放音检测。

[0101] 在本步骤中，当处理器检测到AudioTrack的对象被创建后，则表明应用开始进行放音，此时，处理器可以检测应用的放音事件并获取放音事件对应的放音特征，进而确定

应用是 否处于无效反应状态。

[0102] 示例性的,应用进行放音时时,上层应用会构造一个AudioTrack的对象,并相应的通知 AudioFlinger.AudioFlinger在接收到AudioTrack的对象的创建的通知后,可以确定该应用正 在进行放音。因此,若处理器检测到AudioTrack的对象被创建,则可以确定应用开始执行放 音任务,进而处理器可以开始对应用的无效识别进行检测。

[0103] 在一些实施例中,当处理器检测到AudioTrack的对象被创建后,还可以记录该应用的放 音事件的相关信息。

[0104] 示例性的,应用的放音事件的信息可以包括pid,uid,sessionid和mId。其中,pid为操 作系统中各进程的标识。Uid为用户身份证明。Sessionid为会话控制标识。mId为移动 设备 标识。

[0105] S502、检测应用对应的音轨AudioTrack的音量设置接口,获取应用的设置音量。

[0106] S503、若应用的设置音量小于第一阈值且应用的系统设置音量大于第二阈值,则 确定应 用处于无效放音状态。

[0107] S504、对应用的放音行为进行管控。

[0108] 本实施例中,步骤S502-S504的具体实现过程和实现原理与实施例一中步骤S201-S203 的类似,此处不再赘述。

[0109] S505、若检测到AudioTrack的对象被移除或销毁,则停止应用的无效放音检测。

[0110] 在本步骤中,当应用结束放音时,AudioTrack的对象被移除或销,此时, AudioFlinger 会有接收到线程移除的通知。进而,当检测到应用移除第一线程对象,表明 应用停止进行放 音,进而,可以停止对应用进行无效放音识别。

[0111] 一种可选的实施方式中,AudioTrack的对象被移除或销毁后,还可以记录应用的 放音事 件的信息。

[0112] 示例性的,应用的放音事件的信息可以包括pid,uid,sessionid和mId。其中,pid为操 作系统中各进程的标识。Uid为用户身份证明。Sessionid为会话控制标识。mId为移动 设备 标识。

[0113] 本申请实施例提供的应用的处理方法,若检测到AudioTrack的对象被创建,则启 动应用 的无效放音检测,若检测到AudioTrack的对象被移除或销毁,则停止应用的无效放 音检测, 使得只有在应用放音时检测应用的放音状态,避免了应用不在放音时仍然进行检 测,进而降 低了能耗。

[0114] 需要说明的是,本申请实施例中可以仅采用实施例一、实施例二、实施例三和实 施例 四中的一种无效放音识别方式来进行无效放音识别,也可以采用实施例一、实施例二、 实施例三 和实施例四的一种或多种方式相互结合来进行无效放音识别。示例性的,可 以先 采用应用的设置音量进行无效放音识别,若未检测到无效放音,再采用实际播放音 量进 行无效放音识别,若仍

[0115] 图8为本申请实施例提供的一种应用的管控装置示意图。如图8,当应用进行放音 时, 终端设备的状态识别装置会从AudioTrack中识别静音音频帧的数量和应用设置音量, 从 AudioFlinger中识别实际播放音量,从而确定应用是否处于无效放音状态。若状态识别 装置确 定应用处于无效放音状态,可以将该信息发送给管控装置,从而使管控装置停止无 效放音应 用的放音事件。

[0116] 其中,方式1可以为实施例一种所记载的应用的处理方法,方式2可以为实施例二种所记载的应用的处理方法,方式3可以为实施例三种所记载的应用的处理方法,方式4可以为实施例四中步骤S501中所记载的应用的处理方法,方式5可以为实施例四中步骤S505中所记载的应用的处理方法。在AudioTrack中可以采用方式1-4中的一种或多种进行应用的处理,在AudioFlinger可以采用方式5进行应用的处理。

[0117] 下面结合图9对本申请实施例提供的应用的处理方法中如何停止应用的无效放音进行详细说明。图9为本申请实施例提供的应用的处理方法的实施例五的流程示意图,如图9所示,本实施例提供的应用的处理方法可以包括:

[0118] S601、检测应用对应的音轨AudioTrack的音量设置接口,获取应用的设置音量。

[0119] S602、若应用的设置音量小于第一阈值且应用的系统设置音量大于第二阈值,则确定应用处于无效放音状态。

[0120] 本实施例中,步骤S601-S602的具体实现过程和实现原理与实施例一中步骤S201-S202的类似,此处不再赘述。

[0121] S603、关闭应用的放音进程。

[0122] 在本申请实施例中,可以采用通过三种管控方式分级对应用的无效放音进行管控,从而停止无效放音应用的放音事件,以防恶意应用重复进行无效放音。

[0123] 其中,一级管控方式可以为关闭应用的放音进程。当应用第一次进行无效放音时,可以通过一级管控停止应用的放音事件。通过关闭应用的放音进程停止应用的无效放音,对应用和系统影响最小,系统检测到应用的进程停止后,会释放相应的音频资源,降低不必要的功耗。

[0124] S604、若再次检测到应用处于无效放音状态,则关闭应用的前台界面并停止应用的后台服务。

[0125] 本步骤中,恶意应用可通过守护进程监控被关闭的放音进程,然后关联拉起该应用的放音进程,继续进行无效放音。若关闭应用的放音进程后,再次检测到应用处于无效放音状态,则可以采用二级管控方式,通过关闭应用的前台界面并停止应用的后台服务停止应用的放音行为。

[0126] S605、若再次检测到应用处于无效放音状态,则强制停用应用。

[0127] 在本申请实施例中,若然而极少数恶意的应用在关闭应用的前台界面并停止应用的后台服务后,仍然通过JobScheduler、Alarm或网络通知等方式再次启动应用进行恶意放音。此时,可以采用三级管控方式通过强制停用应用来停止应用的放音行为。被强制停止的应用将被完全管控,除非用户主动点击图标在前台启动应用,否则无法通过任何方式在后台自动启动。

[0128] 本申请实施例提供的应用的处理方法,可以通过关闭应用的放音进程,对应用的放音行为进行管控,若再次检测到应用处于无效放音状态,则关闭应用的前台界面并停止应用的后台服务;若再次检测到应用处于无效放音状态,则强制停用应用。通过该方式,采用逐级管控的方式,确定停止应用的无效放音。

[0129] 图10为本申请实施例提供的一种应用的处理装置。该应用处理的终端设备的处理装置可以通过软件、硬件或者两者的结合实现。该应用处理的终端设备的处理装置可以包括:处理器71和存储器72,其中,

[0130] 存储器71用于存储程序;

[0131] 处理器72用于执行程序,以检测应用对应的音轨AudioTrack的音量设置接口,获取应用的设置音量;若应用的设置音量小于第一阈值且应用的系统设置音量大于第二阈值,则确定应用处于无效放音状态;对应用的放音行为进行管控。

[0132] 一种可选的实施方式中,处理器72还可以执行程序,以检测应用对应的AudioFlinger的设置最终音量接口,获取应用的实际播放音量;若应用的实际播放音量小于第三阈值且应用的系统实际音量大于第四阈值,则确定应用处于无效放音状态。

[0133] 一种可选的实施方式中,处理器72还可以执行程序,以获取AudioTrack连续调用write接口输入静音音频帧的数量;若静音音频帧的数量大于第五阈值,则确定应用处于无效放音状态。

[0134] 一种可选的实施方式中,处理器72还可以执行程序,以关闭应用的放音进程。

[0135] 一种可选的实施方式中,处理器72还可以执行程序,以使得若再次检测到应用处于无效放音状态,则关闭应用的前台界面并停止应用的后台服务。

[0136] 一种可选的实施方式中,处理器72还可以执行程序,以使得若再次检测到应用处于无效放音状态,则强制停用应用。

[0137] 一种可选的实施方式中,处理器72还可以执行程序,以在检测应用对应的音轨AudioTrack的音量设置接口,获取应用的设置音量之前,若检测到AudioTrack的对象被创建,则启动应用的无效放音检测。

[0138] 一种可选的实施方式中,处理器72还可以执行程序,以使得若检测到AudioTrack的对象被移除或销毁,则停止应用的无效放音检测。

[0139] 图11为本申请实施例提供的一种电子设备的结构示意图。如图9所示,该电子设备可以包括:处理器81(例如CPU)、存储器82、收发器83;收发器83耦合至处理器81,处理器81控制收发器83的收发动作;存储器82可能包含高速随机存取存储器(random-access memory, RAM),也可能还包括非易失性存储器(non-volatile memory, NVM),例如至少一个磁盘存储器,存储器82中可以存储各种指令,以用于完成各种处理功能以及实现本申请的方法步骤。在一种可实施的方式中,本申请涉及的电子设备还可以包括:电源84、通信总线85以及通信端口86。收发器83可以集成在应用的处理装置的收发信机中,也可以为应用的处理装置上独立的收发天线。通信总线85用于实现元件之间的通信连接。上述通信端口86用于实现终端设备与其他外设之间进行连接通信。

[0140] 在本申请实施例中,上述存储器82用于存储计算机可执行程序代码,程序代码包括指令;当处理器81执行指令时,指令使终端设备的处理器81执行上述方法实施例中应用的处理装置的处理动作,使收发器83执行上述方法实施例中应用的处理装置的收发动作,其实现原理和技术效果类似,在此不再赘述。

[0141] 正如上述实施例,本申请实施例涉及的应用的处理装置可以是手机、平板电脑等无线终端,因此,以应用的处理装置为手机为例:图12为本申请实施例提供的应用的处理装置为手机时的结构框图。参考图12,该手机可以包括:射频(Radio Frequency, RF)电路1110、存储器1120、输入单元1130、显示单元1140、传感器1150、音频电路1160、无线保真(wireless fidelity, WiFi)模块1170、处理器1180、以及电源1190等部件。本领域技术人员可以理解,图12中示出的手机结构并不构成对手机的限定,可以包括比图示更多或更少的

部件,或者组合某些部件,或者不同的部件布置。

[0142] 下面结合图12对手机的各个构成部件进行具体的介绍:

[0143] RF电路1110可用于收发信息或通话过程中,信号的接收和发送,例如,将基站的下行信息接收后,给处理器1180处理;另外,将上行的数据发送给基站。通常,RF电路包括但不限于天线、至少一个放大器、收发信机、耦合器、低噪声放大器(Low Noise Amplifier, LNA)、双工器等。此外,RF电路1110还可以通过无线通信与网络和其他设备通信。上述无线通信可以使用任一通信标准或协议,包括但不限于全球移动通讯系统(Global System of Mobile communication,GSM)、通用分组无线服务(General Packet Radio Service, GPRS)、码分多址(Code Division Multiple Access,CDMA)、宽带码分多址(Wideband Code Division Multiple Access,WCDMA)、长期演进(Long Term Evolution,LTE)、电子邮件、短消息服务(Short Messaging Service,SMS)等。

[0144] 存储器1120可用于存储软件程序以及模块,处理器1180通过运行存储在存储器1120的软件程序以及模块,从而执行手机的各种功能应用以及数据处理。存储器1120可主要包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需的应用程序(比如声音播放功能、图像播放功能等)等;存储数据区可存储根据手机的使用所创建的数据(比如音频数据、电话本等)等。此外,存储器1120可以包括高速随机存取存储器,还可以包括非易失性存储器,例如至少一个磁盘存储器件、闪存器件、或其他易失性固态存储器件。

[0145] 输入单元1130可用于接收输入的数字或字符信息,以及产生与手机的用户设置以及功能控制有关的键信号输入。具体地,输入单元1130可包括触控面板1131以及其他输入设备1132。触控面板1131,也称为触摸屏,可收集用户在其上或附近的触摸操作(比如用户使用手指、触笔等任何适合的物体或附件在触控面板1131上或在触控面板1131附近的操作),并根据预先设定的程式驱动相应的连接装置。可选的,触控面板1131可包括触摸检测装置和触摸控制器两个部分。其中,触摸检测装置检测用户的触摸方位,并检测触摸操作带来的信号,将信号传送给触摸控制器;触摸控制器从触摸检测装置上接收触摸信息,并将它转换成触点坐标,再送给处理器1180,并能接收处理器1180发来的命令并加以执行。此外,可以采用电阻式、电容式、红外线以及表面声波等多种类型实现触控面板1131。除了触控面板1131,输入单元1130还可以包括其他输入设备1132。具体地,其他输入设备1132可以包括但不限于物理键盘、功能键(比如音量控制按键、开关按键等)、轨迹球、鼠标、操作杆等中的一种或多种。

[0146] 显示单元1140可用于显示由用户输入的信息或提供给用户的信息以及手机的各种菜单。显示单元1140可包括显示面板1141,可选的,可以采用液晶显示器(Liquid Crystal Display, LCD)、有机发光二极管(Organic Light-Emitting Diode,OLED)等形式来配置显示面板1141。进一步的,触控面板1131可覆盖于显示面板1141之上,当触控面板1131检测到在其上或附近的触摸操作后,传送给处理器1180以确定触摸事件的类型,随后处理器1180根据触摸事件的类型在显示面板1141上提供相应的视觉输出。虽然在图10中,触控面板1131与显示面板1141是作为两个独立的部件来实现手机的输入和输入功能,但是在某些实施例中,可以将触控面板1131与显示面板1141集成而实现手机的输入和输出功能。

[0147] 手机还可包括至少一种传感器1150,比如光传感器、运动传感器以及其他传感器。具体地,光传感器可包括环境光传感器及接近传感器,其中,环境光传感器可根据环境光线的明暗来调节显示面板1141的亮度,光传感器可在手机移动到耳边时,关闭显示面板1141和/或背光。作为运动传感器的一种,加速度传感器可检测各个方向上(一般为三轴)加速度的大小,静止时可检测出重力的大小及方向,可用于识别手机姿态的应用(比如横竖屏切换、相关游戏、磁力计姿态校准)、振动识别相关功能(比如计步器、敲击)等;至于手机还可配置的陀螺仪、气压计、湿度计、温度计、红外线传感器等其他传感器,在此不再赘述。

[0148] 音频电路1160、扬声器1161以及传声器1162可提供用户与手机之间的音频接口。音频电路1160可将接收到的音频数据转换后的电信号,传输到扬声器1161,由扬声器1161转换为声音信号输出;另一方面,传声器1162将收集的声音信号转换为电信号,由音频电路1160接收后转换为音频数据,再将音频数据输出处理器1180处理后,经RF电路1110以发送给比如另一手机,或者将音频数据输出至存储器1120以便进一步处理。

[0149] WiFi属于短距离无线传输技术,手机通过WiFi模块1170可以帮助用户收发电子邮件、浏览网页和访问流式媒体等,它为用户提供了无线的宽带互联网访问。虽然图12示出了WiFi模块1170,但是可以理解的是,其并不属于手机的必须构成,完全可以根据需要在不改变本申请实施例的本质的范围内而省略。

[0150] 处理器1180是手机的控制中心,利用各种接口和线路连接整个手机的各个部分,通过运行或执行存储在存储器1120内的软件程序和/或模块,以及调用存储在存储器1120内的数据,执行手机的各种功能和处理数据,从而对手机进行整体监控。可选的,处理器1180可包括一个或多个处理单元;例如,处理器1180可集成应用处理器和调制解调处理器,其中,应用处理器主要处理操作系统、用户界面和应用程序等,调制解调处理器主要处理无线通信。可以理解的是,上述调制解调处理器也可以不集成到处理器1180中。

[0151] 手机还包括给各个部件供电的电源1190(比如电池),可选的,电源可以通过电源管理系统与处理器1180逻辑相连,从而通过电源管理系统实现管理充电、放电、以及功耗管理等功能。

[0152] 手机还可以包括摄像头1200,该摄像头可以为前置摄像头,也可以为后置摄像头。尽管未示出,手机还可以包括蓝牙模块、GPS模块等,在此不再赘述。

[0153] 在本申请实施例中,该手机所包括的处理器1180可以用于执行上述数据传输方法实施例,其实现原理和技术效果类似,在此不再赘述。

[0154] 本申请实施例还提供了一种芯片,包括处理器和接口。其中接口用于输入输出处理器所处理的数据或指令。处理器用于执行以上方法实施例中提供的方法。该芯片可以应用于终端设备中。

[0155] 本申请实施例还提供一种程序,该程序在被处理器执行时用于执行以上方法实施例提供的方法。

[0156] 本申请实施例还提供一种程序产品,例如计算机可读存储介质,该程序产品中存储有指令,当其在计算机上运行时,使得计算机执行上述方法实施例提供的方法。

[0157] 在上述实施例中,可以全部或部分地通过软件、硬件、固件或者其任意组合来实现。当使用软件实现时,可以全部或部分地以计算机程序产品的形式实现。计算机程序产

品包括一个或多个计算机指令。在计算机上加载和执行计算机程序指令时,全部或部分地产生按照本发明实施例的流程或功能。计算机可以是通用计算机、专用计算机、计算机网络、或者其他可编程装置。计算机指令可以存储在计算机可读存储介质中,或者从一个计算机可读存储介质向另一个计算机可读存储介质传输,例如,计算机指令可以从一个网站站点、计算机、服务器或数据中心通过有线(例如同轴电缆、光纤、数字用户线(DSL))或无线(例如红外、无线、微波等)方式向另一个网站站点、计算机、服务器或数据中心进行传输。计算机可读存储介质可以是计算机能够存取的任何可用介质或者是包含一个或多个可用介质集成的服务器、数据中心等数据存储设备。可用介质可以是磁性介质,(例如,软盘、硬盘、磁带)、光介质(例如,DVD)、或者半导体介质(例如固态硬盘Solid State Disk(SSD))等。

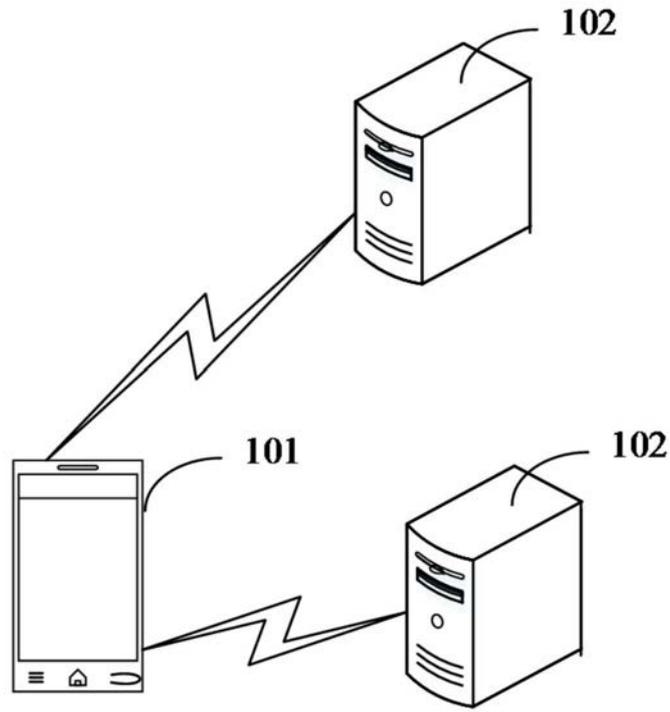


图1

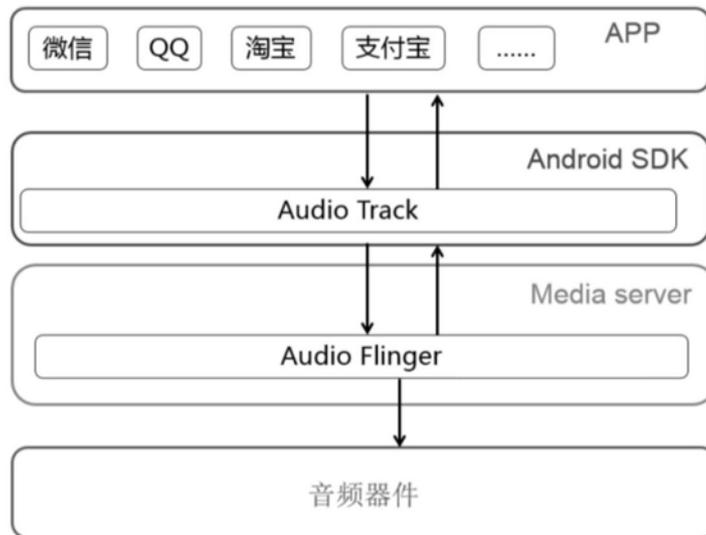


图2



图3



图4

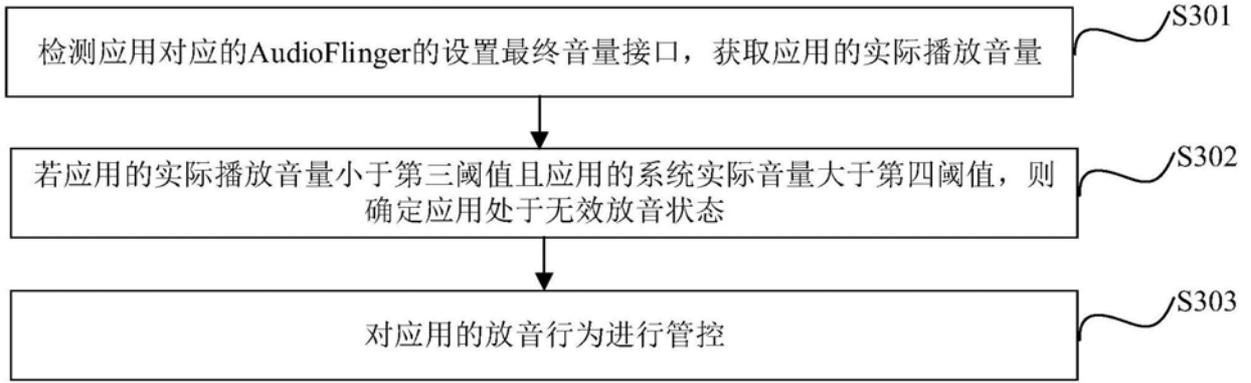


图5

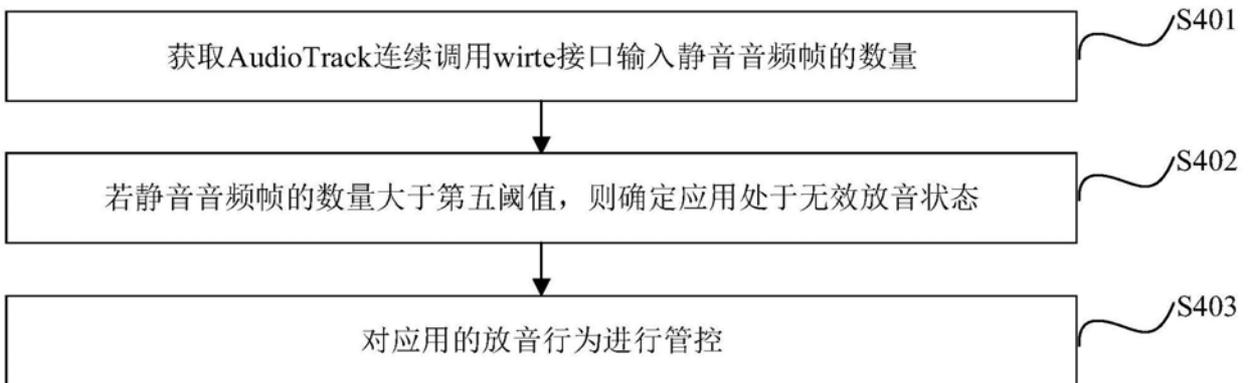


图6

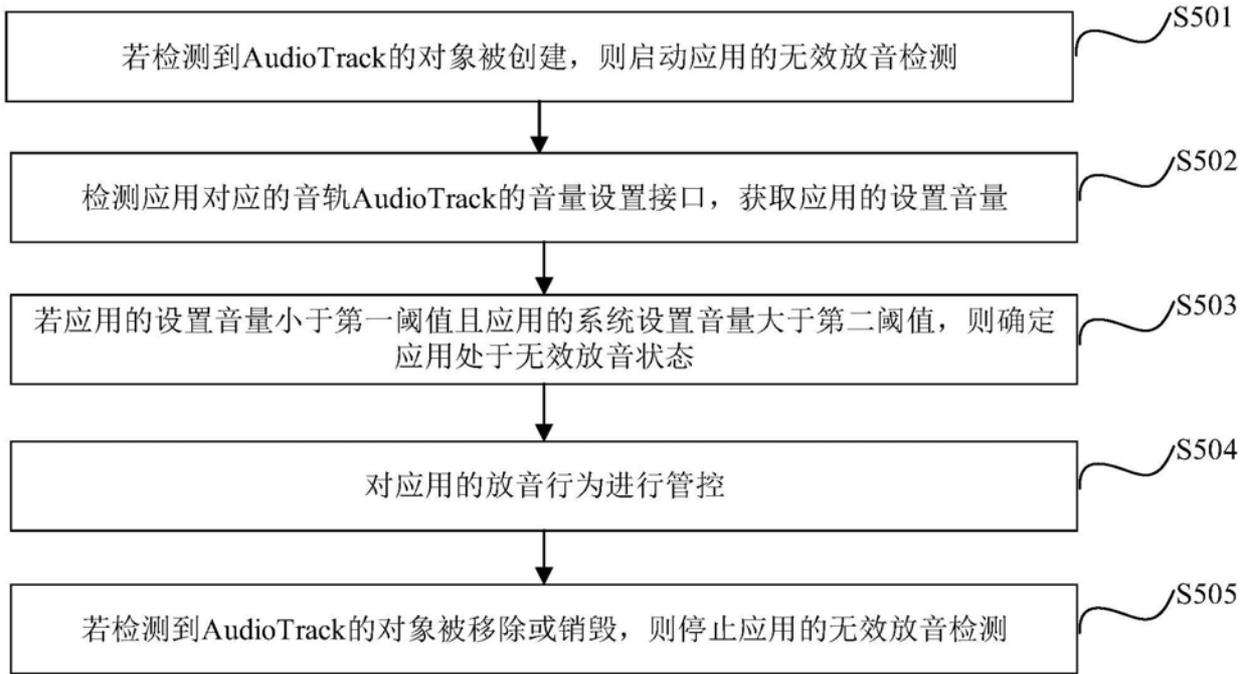


图7

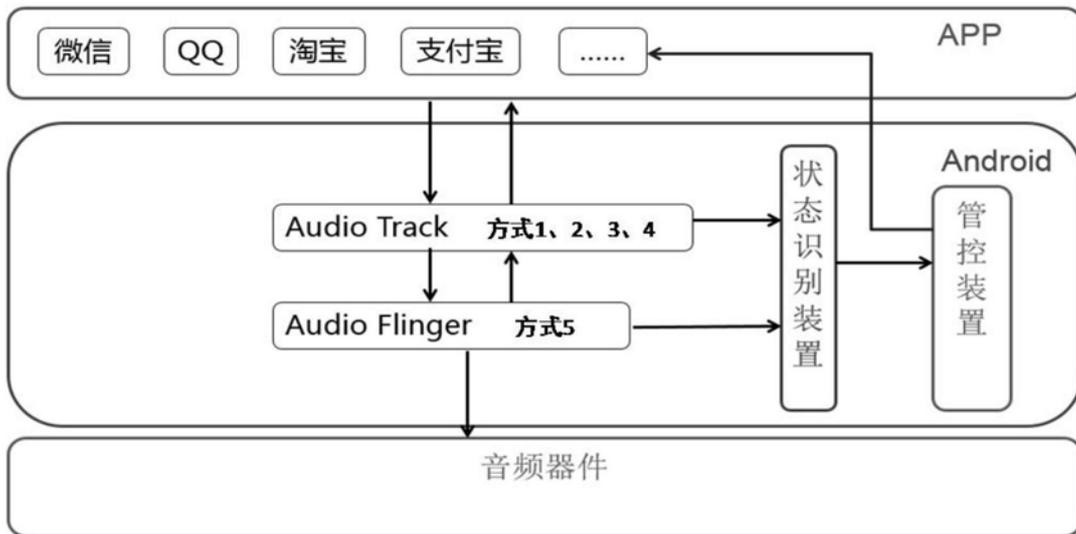


图8

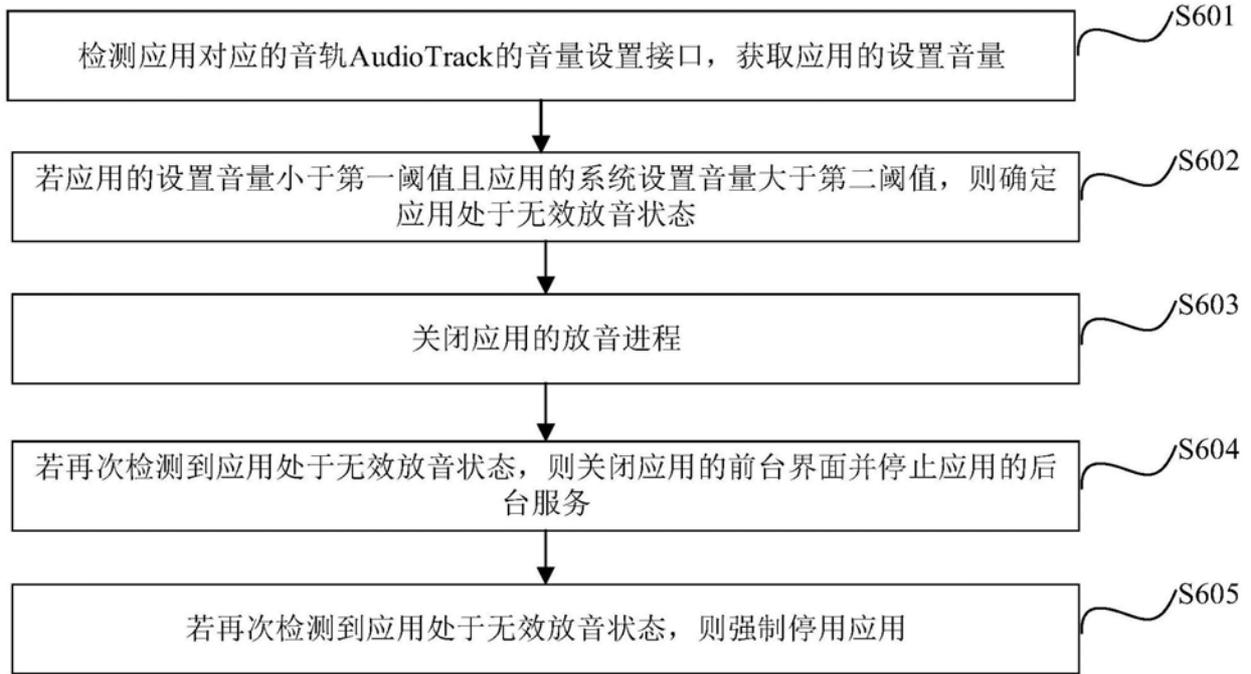


图9

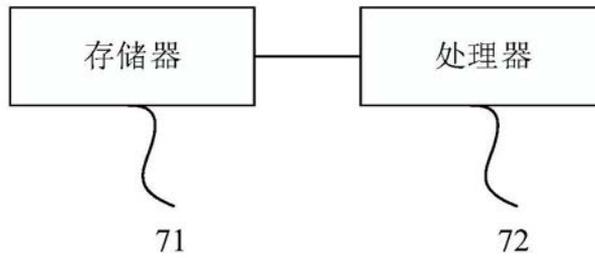


图10

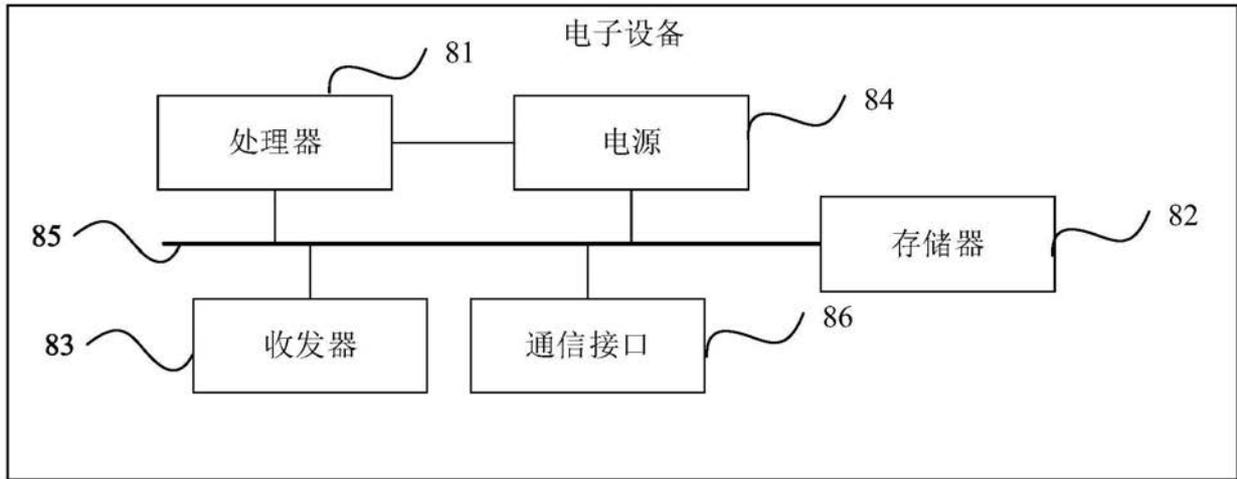


图11

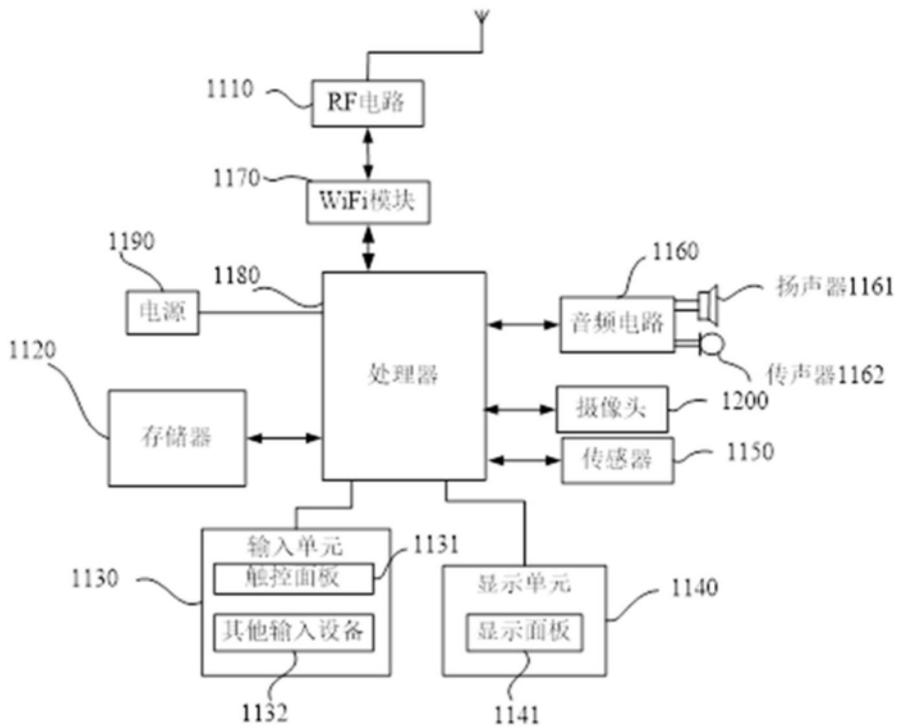


图12