



(12) 发明专利申请

(10) 申请公布号 CN 103309668 A

(43) 申请公布日 2013. 09. 18

(21) 申请号 201310239696. 9

(22) 申请日 2013. 06. 17

(71) 申请人 深圳TCL新技术有限公司

地址 518052 广东省深圳市南山区中山园路
1001号 TCL国际E城科技大厦D4栋7
楼

(72) 发明人 罗朝江

(74) 专利代理机构 深圳市世纪恒程知识产权代
理事务所 44287

代理人 胡海国

(51) Int. Cl.

G06F 9/44 (2006. 01)

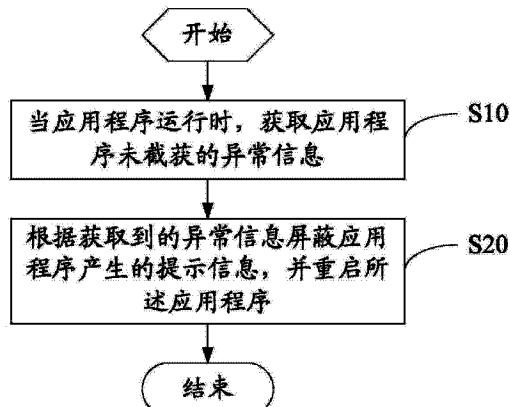
权利要求书2页 说明书6页 附图4页

(54) 发明名称

应用程序未截获异常的处理方法及装置

(57) 摘要

本发明公开了一种应用程序未截获异常的处理方法及装置，所述应用程序未截获异常的处理包括以下步骤：应用出现未截获异常时，系统中截获应用程序未截获的异常信息，根据获取到的异常信息屏蔽应用程序产生的提示信息，并重启所述应用程序。本发明降低了因应用程序运行出错而出现错误提示对话框给用户带来的繁琐操作，从而提高了用户的体验度。



1. 一种应用程序未截获异常的处理方法,其特征在于,包括以下步骤:

A、当应用程序运行时,获取应用程序未截获的异常信息;

B、根据获取到的异常信息屏蔽应用程序产生的提示信息,并重启所述应用程序。

2. 如权利要求1所述的应用程序未截获异常的处理方法,其特征在于,在执行所述步骤B之前还包括:

C、判断所述应用程序重启的次数是否大于预置值;若是则执行步骤D,否则执行步骤B;

D、根据获取到的异常信息屏蔽应用程序产生的提示信息,并结束当前应用程序的启动。

3. 如权利要求1所述的应用程序未截获异常的处理方法,其特征在于,在执行所述步骤B之后还包括:

E、清空重启所述应用程序前所述应用程序所占用的资源。

4. 如权利要求3所述的应用程序未截获异常的处理方法,其特征在于,在执行所述步骤B之前还包括:

F、判断启动应用程序至获取到异常信息之间的时间差是否小于预置时间;若是则执行步骤G,否则执行步骤B;

G、等待所述应用程序启动时间达到预置时间时,执行所述步骤B。

5. 如权利要求1所述的应用程序未截获异常的处理方法,其特征在于,所述步骤A之前包括:

通过Android的commonInit机制配置异常处理程序,并在所述异常处理程序中,通过调用handleApplicationCrash函数进行应用程序异常处理。

6. 一种应用程序未截获异常的处理装置,其特征在于,包括

获取模块,用于当应用程序运行时,获取应用程序未截获的异常信息;

处理模块,用于根据获取到的异常信息屏蔽应用程序产生的提示信息,并重启所述应用程序。

7. 如权利要求6所述的应用程序未截获异常的处理装置,其特征在于,所述应用程序未截获异常的处理装置还包括:

第一判断模块,用于判断所述应用程序重启的次数是否大于预置值;

所述处理模块具体用于,当所述应用程序重启的次数大于预置值时,根据获取到的异常信息屏蔽应用程序产生的提示信息,并结束当前应用程序的启动;当所述应用程序重启的次数小于等于预置值时,执行所述根据获取到的异常信息屏蔽应用程序产生的提示信息,并重启所述应用程序的操作。

8. 如权利要求6所述的应用程序未截获异常的处理装置,其特征在于,所述应用程序未截获异常的处理装置还包括:

清空模块,用于清空重启所述应用程序前所述应用程序所占用的资源。

9. 如权利要求8所述的应用程序未截获异常的处理装置,其特征在于,所述应用程序未截获异常的处理装置还包括:

第二判断模块,用于判断启动应用程序至获取到异常信息之间的时间差是否小于预置时间;

所述处理模块具体用于，当所述时间差小于预置时间时，等待所述应用程序启动时间达到预置时间时，执行所述根据获取到的异常信息屏蔽应用程序产生的提示信息，并重启所述应用程序的操作；当所述时间差大于等于预置时间时，直接执行所述根据获取到的异常信息屏蔽应用程序产生的提示信息，并重启所述应用程序的操作。

10. 如权利要求 6 所述的应用程序未截获异常的处理装置，其特征在于，

还包括未截获异常处理模块，用于通过 Android 的 commonInit 机制配置异常处理程序，并在所述异常处理程序中，通过调用 handleApplicationCrash 函数进行应用程序异常处理。

应用程序未截获异常的处理方法及装置

技术领域

[0001] 本发明涉及显示终端领域,特别涉及一种应用程序未截获异常的处理方法及装置。

背景技术

[0002] 众所周知,在Android内存溢出时,会抛出OOM(Out Of Memory,内存不足)异常,此异常在应用程序的代码中是不能截获的。此类未截获的异常发生时,系统会弹出一个错误提示对话框,此时用户需要关闭提示对话框并重新点击运行此应用程序,由于此种方式的操作繁琐,从而使得用户的体验度较差。

发明内容

[0003] 本发明的主要目的在于提供一种应用程序未截获异常的处理方法,旨在降低因应用程序运行出错而出现错误提示对话框给用户带来的繁琐操作,以提高用户的体验度。

[0004] 为了实现发明目的,本发明提供一种应用程序未截获异常的处理方法,该应用程序未截获异常的处理方法包括以下步骤:

[0005] A、当应用程序运行时,获取应用程序未截获的异常信息;

[0006] B、根据获取到的异常信息屏蔽应用程序产生的提示信息,并重启所述应用程序。

[0007] 优选地,在执行所述步骤B之前还包括:

[0008] C、判断所述应用程序重启的次数是否大于预置值;若是则执行步骤D,否则执行步骤B;

[0009] D、根据获取到的异常信息屏蔽应用程序产生的提示信息,并结束当前应用程序的启动。

[0010] 优选地,在执行所述步骤B之后还包括:

[0011] E、清空重启所述应用程序前所述应用程序所占用的资源。

[0012] 优选地,在执行所述步骤B之前还包括:

[0013] F、判断启动应用程序至获取到异常信息之间的时间差是否小于预置时间;若是则执行步骤G,否则执行步骤B;

[0014] G、等待所述应用程序启动时间达到预置时间时,执行所述步骤B。

[0015] 优选地,所述步骤A之前包括:

[0016] 通过Android的commonInit机制配置异常处理程序,并在所述异常处理程序中,通过调用handleApplicationCrash函数进行应用程序异常处理。

[0017] 本发明还提供一种应用程序未截获异常的处理装置,该应用程序未截获异常的处理装置包括

[0018] 获取模块,用于当应用程序运行时,获取应用程序未截获的异常信息;

[0019] 处理模块,用于根据获取到的异常信息屏蔽应用程序产生的提示信息,并重启所述应用程序。

- [0020] 优选地，所述应用程序未截获异常的处理装置还包括：
- [0021] 第一判断模块，用于判断所述应用程序重启的次数是否大于预置值；
- [0022] 所述处理模块具体用于，当所述应用程序重启的次数大于预置值时，根据获取到的异常信息屏蔽应用程序产生的提示信息，并结束当前应用程序的启动；当所述应用程序重启的次数小于等于预置值时，执行所述根据获取到的异常信息屏蔽应用程序产生的提示信息，并重启所述应用程序的操作。
- [0023] 优选地，所述应用程序未截获异常的处理装置还包括：
- [0024] 清空模块，用于清空重启所述应用程序前所述应用程序所占用的资源。
- [0025] 优选地，所述应用程序未截获异常的处理装置还包括：
- [0026] 第二判断模块，用于判断启动应用程序至获取到异常信息之间的时间差是否小于预置时间；
- [0027] 所述处理模块具体用于，当所述时间差小于预置时间时，等待所述应用程序启动时间达到预置时间时，执行所述根据获取到的异常信息屏蔽应用程序产生的提示信息，并重启所述应用程序的操作；当所述时间差大于等于预置时间时，直接执行所述根据获取到的异常信息屏蔽应用程序产生的提示信息，并重启所述应用程序的操作。
- [0028] 优选地，所述应用程序未截获异常的处理装置还包括未截获异常处理模块，用于通过 Android 的 commonInit 机制配置异常处理程序，并在所述异常处理程序中，通过调用 handleApplicationCrash 函数进行应用程序异常处理。
- [0029] 本发明通过获取应用程序未截获的异常信息，并根据该异常信息屏蔽应用程序产生的提示信息，同时通过控制应用程序自动重启。因此降低了因应用程序运行出错而出现错误提示对话框给用户带来的繁琐操作，同时有效保证了应用程序启动过程中不存在间断，从而有效提高了用户的体验度。

附图说明

- [0030] 图 1 为本发明应用程序未截获异常的处理方法第一实施例的流程示意图；
- [0031] 图 2 为本发明应用程序未截获异常的处理方法第二实施例的流程示意图；
- [0032] 图 3 为本发明应用程序未截获异常的处理方法第三实施例的流程示意图；
- [0033] 图 4 为本发明应用程序未截获异常的处理方法第四实施例的流程示意图；
- [0034] 图 5 为本发明应用程序未截获异常的处理装置一实施例的结构示意图。
- [0035] 本发明目的的实现、功能特点及优点将结合实施例，参照附图做进一步说明。

具体实施方式

- [0036] 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] 本发明提供一种应用程序未截获异常的处理方法。
- [0038] 参照图 1，图 1 为本发明应用程序未截获异常的处理方法第一实施例的流程示意图。本实施例提供的应用程序未截获异常的处理方法包括以下步骤：
- [0039] 步骤 S10，当应用程序运行时，获取应用程序未截获的异常信息；
- [0040] 本实施例中，上述应用程序为安卓操作系统中的应用程序，Android 平台在应用进程创建后，将为虚拟机设置一未截获异常处理模块。当用户点击相应的应用程序时，相应的

应用程序将被运行,此时若存在应用程序未截获的异常信息时,应用进程将抛出应用程序未截获的异常信息至未截获异常处理模块。例如,当Android内存溢出时,应用进程将抛出OOM异常。值得说明的是,操作系统底层包括基本异常捕获的机制,本方案在于对操作系统的异常捕获机制进行完善。

[0041] 步骤S20,根据获取到的异常信息屏蔽应用程序产生的提示信息,并重启所述应用程序。

[0042] 本实施例中,上述提示信息为应用程序因上述异常信息所产生的提示信息,例如,当Android内存溢出时,将通过对话框的形式提示用户当前运行的内存不足等提示信息。当上述未截获异常处理模块接收到上述异常信息后,将屏蔽应用程序产生的提示信息,并控制应用程序自动重启或做其他处理。

[0043] 在本实施例中,可通过修改Android底层源码,通过Android的commonInit机制配置异常处理程序,并设置在所述异常处理程序中处理应用程序异常。

[0044] 具体地,通过重写其应用程序异常处理程序,调用ActivityManagerService的handleApplicationCrash函数进行异常处理。从而实现在操作系统底层实现应用程序异常的捕捉与处理。

[0045] 本发明通过获取应用程序未截获的异常信息,并根据该异常信息屏蔽应用程序产生的提示信息,同时通过控制应用程序自动重启。因此降低了因应用程序运行出错而出现错误提示对话框给用户带来的繁琐与不友好,同时有效保证了应用程序启动过程中不存在间断,从而有效提高用户的体验度。

[0046] 参照图2,图2为本发明应用程序未截获异常的处理方法第二实施例的流程示意图。本实施例中,在执行上述步骤S20之前还包括:

[0047] 步骤S30,判断所述应用程序重启的次数是否大于预置值;若是则执行步骤S40,否则执行步骤S20;

[0048] 步骤S40,根据获取到的异常信息屏蔽应用程序产生的提示信息,并结束当前应用程序的启动。优选地,在该步骤还可以生成操作系统统一的异常提示信息,并提示用户,或者做其他处理。

[0049] 本实施例中,上述预置值的大小可根据实际需要进行设置,在此不作进一步的限定。以预置值是10为例,通过定义一数值变量,在应用程序每次重启时,该数值变量都加1,通过判断该数值大小是否大于10从而判定应用程序重启的次数是否大于10。当应用程序重启的次数小于等于10时,则将应控制应用程序重启;当应用程序重启的次数大于10时,则退出当前应用程序的启动,返回至上一操作界面(即运行应用程序的前一操作界面,可以为上一应用程序界面,也可以为桌面)。本实施例中,通过对重启的操作次数进行限制,因此可有效避免因应用程序无法正常启动而导致其他应用程序的运行,从而保证了应用程序运行的稳定性。

[0050] 参照图3,图3为本发明应用程序未截获异常的处理方法第三实施例的流程示意图。基于上述实施例,本实施例中,在执行所述步骤S20之后还包括:

[0051] 步骤S50,清空重启所述应用程序前所述应用程序所占用的资源。

[0052] 本实施例中,上述资源包括Activity和Service中的内容。由于对资源进行清空处理,因此可提高应用程序运行的速度,有利于保证操作系统的正常高效运行。

[0053] 参照图 4, 图 4 为本发明应用程序未截获异常的处理方法第四实施例的流程示意图。基于上述实施例, 本实施例中, 在执行所述步骤 S20 之前还包括:

[0054] 步骤 S60, 判断启动应用程序至获取到异常信息之间的时间差是否小于预置时间;若是则执行步骤 S70, 否则执行步骤 S20;

[0055] 步骤 S70, 等待所述应用程序启动时间达到预置时间时, 执行所述步骤 S20。

[0056] 本实施例中, 上述预置时间可根据实际需要进行设置, 以下以预置时间为 1 分钟作出详细说明。在应用程序启动时, 开启计时器对应用程序启动的时间进行计时, 当从应用程序启动开始到上述未截获异常处理模块获取到异常信息的时间差小于 1 分钟时, 则将进行延时等待, 直到应用程序启动时间到达 1 分钟时, 则执行根据获取到的异常信息屏蔽应用程序产生的提示信息, 并重启所述应用程序的操作; 当从应用程序启动开始到上述未截获异常处理模块获取到异常信息的时间差大于等于 1 分钟时, 则直接执行根据获取到的异常信息屏蔽应用程序产生的提示信息, 并重启所述应用程序的操作。

[0057] 本发明还提供一种应用程序未截获异常的处理装置, 用于实现上述方法。参照图 5, 图 5 为本发明应用程序未截获异常的处理装置一实施例的结构示意图。本实施例提供的应用程序未截获异常的处理装置包括:

[0058] 获取模块 100, 用于当应用程序运行时, 获取应用程序未截获的异常信息;

[0059] 本实施例中, 上述应用程序为安卓操作系统中的应用程序, Android 平台在应用进程创建后, 将为虚拟机设置一未截获异常处理模块 600。当用户点击相应的应用程序时, 相应的应用程序将被运行, 此时若存在应用程序未截获的异常信息时, 应用进程将抛出应用程序未截获的异常信息至未截获异常处理模块 600, 由该未截获异常处理模块将该异常信息输出至获取模块 100。例如, 当 Android 内存溢出时, 应用进程将抛出 OOM 异常。值得说明的是, 操作系统底层包括基本异常捕获的机制, 本方案在于对操作系统的异常捕获机制进行完善。

[0060] 处理模块 200, 用于根据获取到的异常信息屏蔽应用程序产生的提示信息, 并重启所述应用程序。

[0061] 本实施例中, 上述提示信息为应用程序因上述异常信息所产生的提示信息, 例如, 当 Android 内存溢出时, 将通过对话框的形式提示用户当前运行的内存不足等提示信息。当上述获取模块 100 获取到上述异常信息后, 将输出一控制信号至处理模块 200, 处理模块 200 将根据该控制信号屏蔽应用程序产生的提示信息, 并控制应用程序自动重启或做其他处理。

[0062] 应当说明的是, 在本实施例中, 可通过修改 Android 底层源码, 通过 Android 的 commonInit 机制配置异常处理程序, 并设置在所述异常处理程序中处理应用程序异常。

[0063] 具体地, 通过重写其应用程序异常处理程序, 调用 ActivityManagerService 的 handleApplicationCrash 函数进行异常处理。从而实现在操作系统底层实现应用程序异常的捕捉与处理。

[0064] 本发明通过获取模块 100 获取应用程序未截获的异常信息, 并根据该异常信息输出控制信号至处理模块 200, 供处理模块 200 根据该控制信号屏蔽应用程序产生的提示信息, 同时通过控制应用程序自动重启。因此防止了因应用程序运行出错而出现错误提示对话框给用户带来的繁琐与不友好, 同时有效保证了应用程序启动过程中不存在间断, 从而

有效提高用户的体验度。

[0065] 进一步地,基于上述实施例,本实施例中,上述应用程序未截获异常的处理装置还包括:

[0066] 第一判断模块 300,用于判断所述应用程序重启的次数是否大于预置值;

[0067] 所述处理模块 200 具体用于,当所述应用程序重启的次数大于预置值时,根据获取到的异常信息屏蔽应用程序产生的提示信息,并结束当前应用程序的启动;当所述应用程序重启的次数小于等于预置值时,执行所述根据获取到的异常信息屏蔽应用程序产生的提示信息,并重启所述应用程序的操作。

[0068] 本实施例中,上述预置值的大小可根据实际需要进行设置,在此不作进一步的限定。以预置值是 10 为例,通过定义一数值变量,在应用程序每次重启时,该数值变量都加 1,上述第一判断模块 300 通过判断该数值大小是否大于 10 从而判定应用程序重启的次数是否大于 10,并将判断的结果输出至处理模块 200,供处理模块 200 执行相应的操作。当应用程序重启的次数小于等于 10 时,则上述处理模块 200 将应用程序重启;当应用程序重启的次数大于 10 时,则退出当前应用程序的启动,返回至上一操作界面(即运行应用程序的前一操作界面,可以为上一应用程序界面,也可以为桌面)。本实施例中,通过对重启的操作次数进行限制,因此可有效避免因应用程序无法正常启动而导致其他应用程序的运行,从而保证了应用程序运行的稳定性。

[0069] 进一步地,基于上述实施例,本实施例中,上述应用程序未截获异常的处理装置还包括:

[0070] 清空模块 400,用于清空重启所述应用程序前所述应用程序所占用的资源。

[0071] 本实施例中,上述资源包括 Activity 和 Service 中的内容。由于对资源进行清空处理,因此可提高应用程序运行的速度,有利于保证应用程序的正常运行。

[0072] 进一步地,基于上述实施例,本实施例中,上述应用程序未截获异常的处理装置还包括:

[0073] 第二判断模块 500,用于判断启动应用程序至获取到异常信息之间的时间差是否小于预置时间;

[0074] 所述处理模块 200 具体用于,当所述时间差小于预置时间时,等待所述应用程序启动时间达到预置时间时,执行所述根据获取到的异常信息屏蔽应用程序产生的提示信息,并重启所述应用程序的操作;当所述时间差大于等于预置时间时,直接执行所述根据获取到的异常信息屏蔽应用程序产生的提示信息,并重启所述应用程序的操作。

[0075] 本实施例中,上述预置时间可根据实际需要进行设置,以下以预置时间为 1 分钟作出详细说明。在应用程序启动时,开启计时器对应用程序启动的时间进行计时,当第二判断模块 500 判定从应用程序启动开始到上述获取模块 100 获取到异常信息的时间差小于 1 分钟时,则输出控制信号至处理模块 200,以供处理模块 200 进行延时等待,直到应用程序启动时间到达 1 分钟时,由处理模块 200 执行根据获取到的异常信息屏蔽应用程序产生的提示信息,并重启所述应用程序的操作;当第二判断模块 500 判定从应用程序启动开始到上述获取模块 100 获取到异常信息的时间差大于等于 1 分钟时,则输出控制信号至处理模块 200,以供处理模块 200 直接执行根据获取到的异常信息屏蔽应用程序产生的提示信息,并重启所述应用程序的操作。

[0076] 以上仅为本发明的优选实施例，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，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。

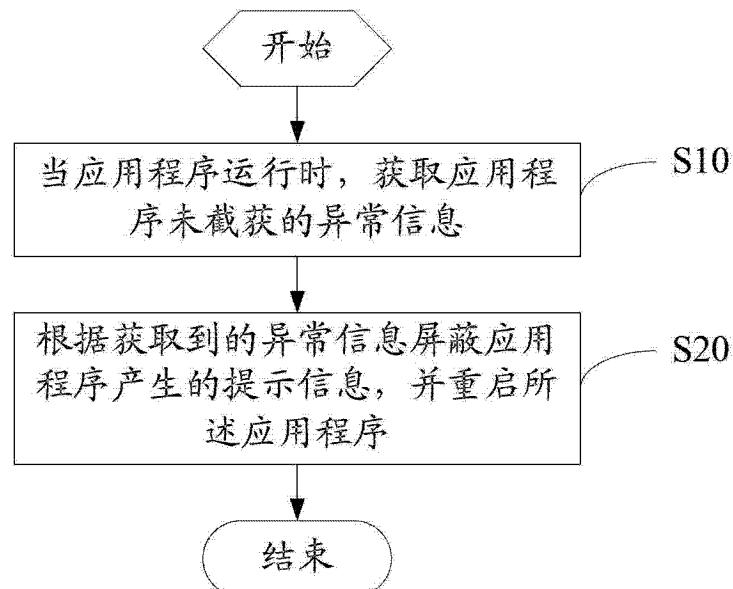


图 1

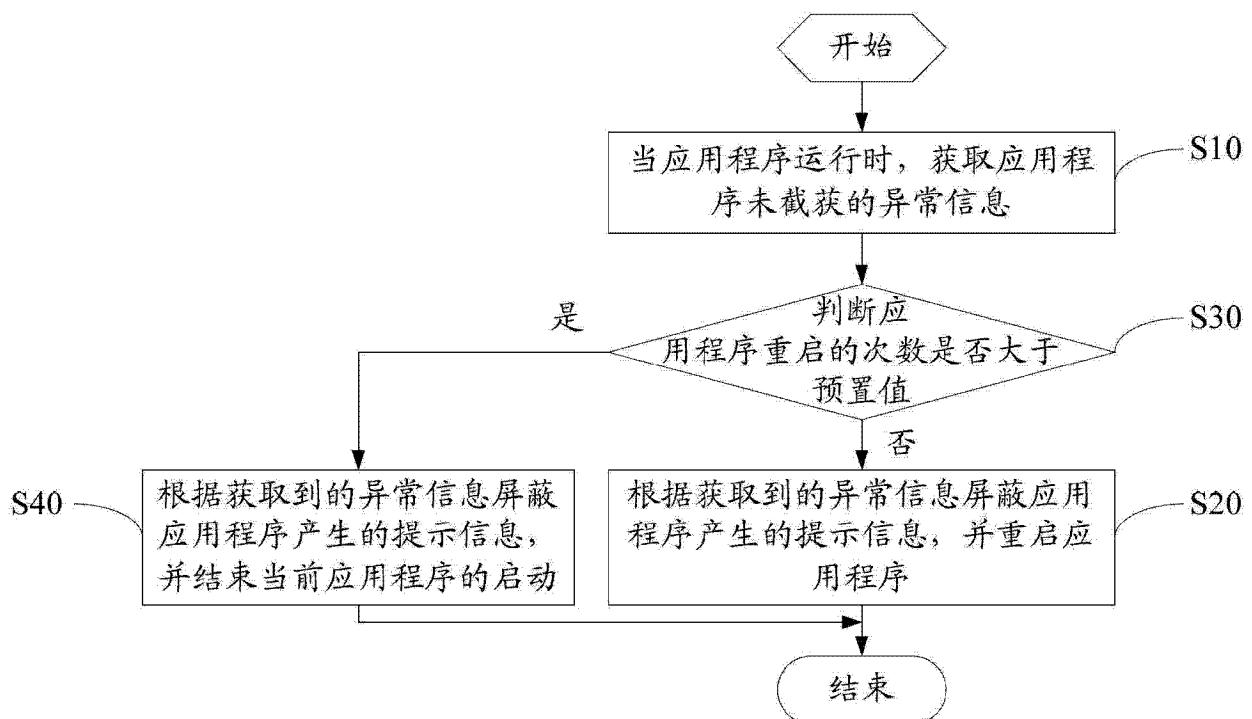


图 2

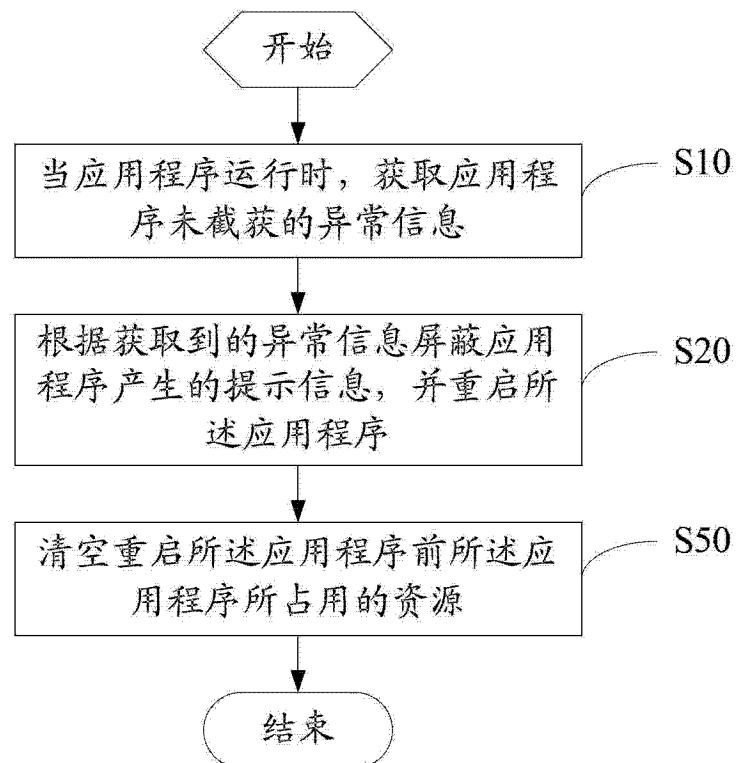


图 3

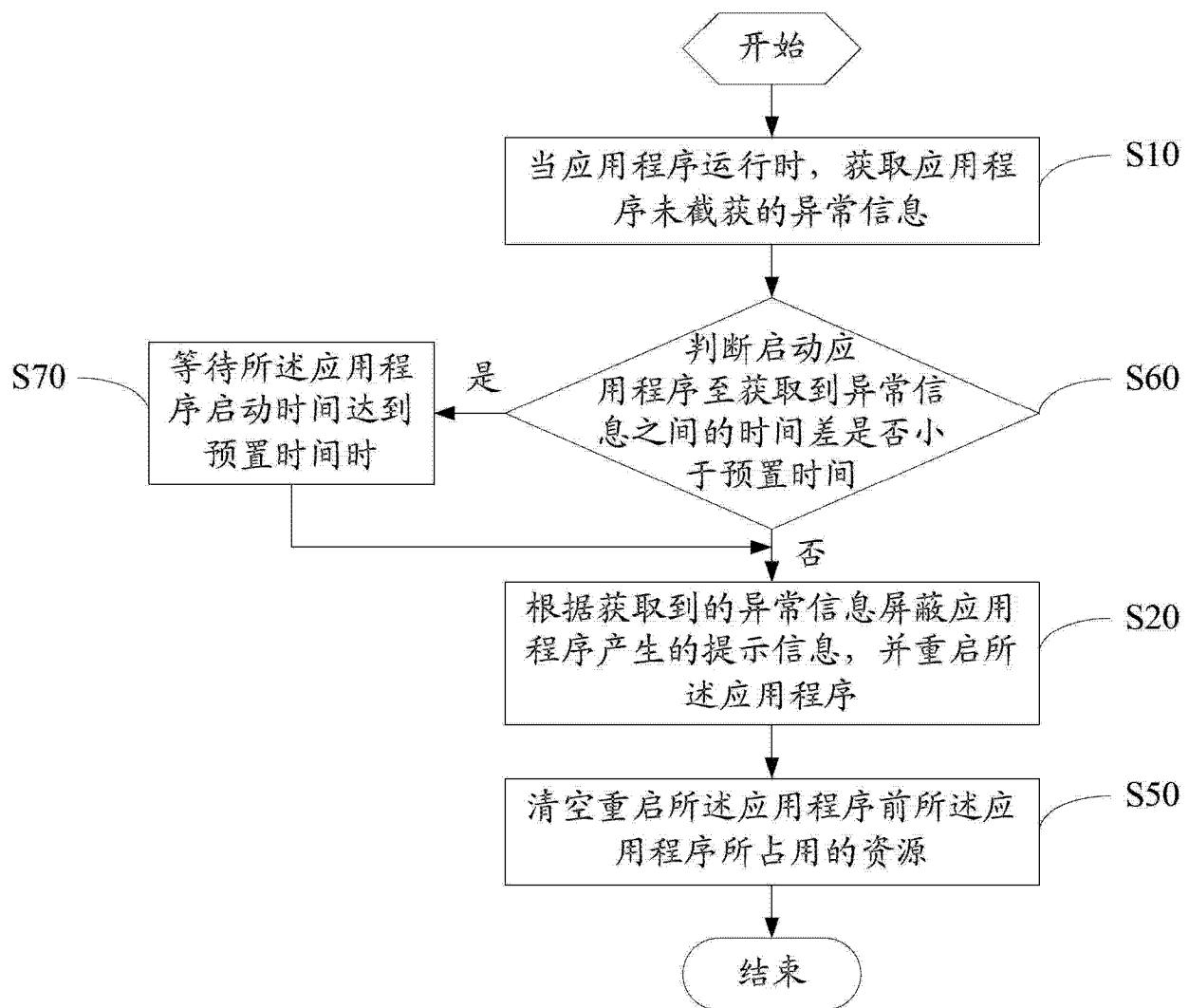


图 4

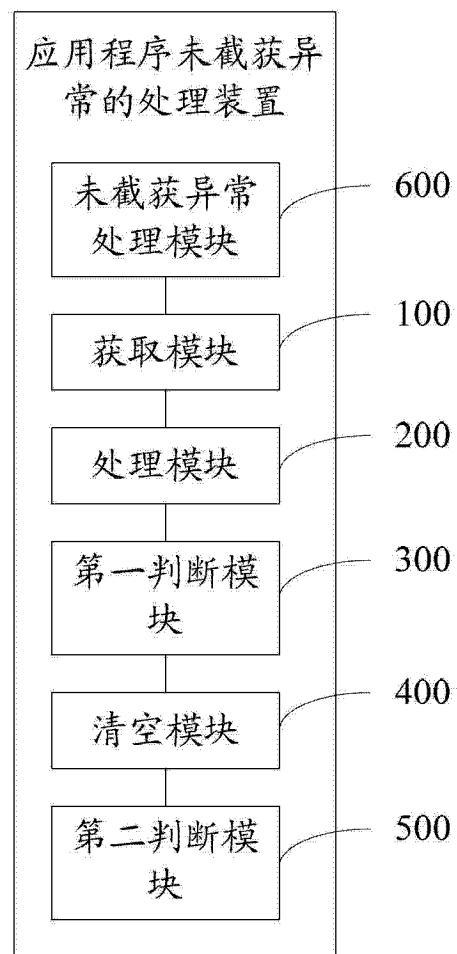


图 5