



(12) 发明专利申请

(10) 申请公布号 CN 113835912 A

(43) 申请公布日 2021.12.24

(21) 申请号 202010584333.9

(22) 申请日 2020.06.24

(71) 申请人 北京新氧科技有限公司
地址 100102 北京市朝阳区创远路34号院8
号楼9层901室、10层1001室

(72) 发明人 李培锋

(74) 专利代理机构 北京中知君达知识产权代理
有限公司 11769
代理人 李辰 黄启法

(51) Int. Cl.
G06F 11/07 (2006.01)

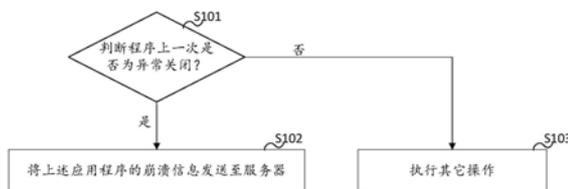
权利要求书2页 说明书8页 附图2页

(54) 发明名称

应用程序的崩溃信息处理方法及设备

(57) 摘要

本申请是关于一种应用程序的崩溃信息处理方法及设备,该方法包括:判断所述应用程序上一次的关闭是否为异常关闭;若是,则将所述应用程序的崩溃信息发送至服务器,其中所述崩溃信息包括崩溃堆栈文件与已经保存的最近N条网络请求的具体信息;其中N为预先确定的正整数。本申请提供的方案,使得服务器端能够快速确定应用程序的崩溃原因。



1. 一种应用程序的崩溃信息处理方法,其特征在于,所述方法包括:
判断所述应用程序的关闭是否为异常关闭;
若是,则将所述应用程序的崩溃信息发送至服务器,所述崩溃信息包括崩溃堆栈文件与已经保存的最近N条网络请求的具体信息;其中N为预先确定的正整数。
2. 根据权利要求1所述的方法,其特征在于,所述方法还包括:
当所述应用程序发送网络请求时,保存所述网络请求的具体信息。
3. 根据权利要求2所述的方法,其特征在于,所述方法还包括:
在所述应用程序初次启动时,在本地创建目标文件,所述目标文件用于保存网络请求的具体信息;
所述保存所述网络请求的具体信息包括:将所述网络请求的具体信息写入所述目标文件。
4. 根据权利要求3所述的方法,其特征在于,所述方法还包括:
按照各网络请求发送时间的先后顺序将多条网络请求的具体信息排列成队列,并保存在所述目标文件中。
5. 根据权利要求4所述的方法,其特征在于,所述保存所述网络请求的具体信息包括:
若所述目标文件中保存的网络请求的具体信息的数量达到第一预定阈值,则删除所述队列中发送时间最早的至少一条网络请求的具体信息,并写入最新的网络请求的具体信息。
6. 根据权利要求4所述的方法,其特征在于,所述保存所述网络请求的具体信息包括:
若所述目标文件中保存的网络请求的具体信息的数量达到第二预定阈值,清空所述队列,并写入最新的网络请求的具体信息。
7. 根据权利要求1所述的方法,其特征在于,所述网络请求的具体信息包括:所述网络请求的发送信息与后端返回信息。
8. 根据权利要求7所述的方法,其特征在于,所述网络请求的发送信息包括下列信息中的至少一种:所述网络请求的统一资源定位符URL、所述网络请求的发送时间、所述网络请求所在页面;
所述网络请求的后端返回信息包括下列信息中的至少一种:所述网络请求是否成功、所述网络请求成功的具体信息、所述网络请求失败的具体信息。
9. 根据权利要求1-8任一项所述的方法,其特征在于,所述判断所述应用程序的关闭是否为异常关闭包括:
当所述应用程序启动时,判断所述应用程序的关闭是否为异常关闭。
10. 一种智能终端,其特征在于,所述终端包括:
判断单元,用于判断所述应用程序上一次的关闭是否为异常关闭;
发送单元,用于当所述判断单元的判断结果为是时,将所述应用程序的崩溃信息发送至服务器,其中所述崩溃信息包括崩溃堆栈文件与已经保存的最近N条网络请求的具体信息;其中N为预先确定的正整数。
11. 一种电子设备,其特征在于,包括:
处理器;以及
存储器,其上存储有可执行代码,当所述可执行代码被所述处理器执行时,使所述处理器执行如权利要求1-9中任一项所述的方法。

12. 一种非暂时性机器可读存储介质,其上存储有可执行代码,当所述可执行代码被电子设备的处理器执行时,使所述处理器执行如权利要求1-9中任一项所述的方法。

应用程序的崩溃信息处理方法及设备

技术领域

[0001] 本申请涉及软件技术领域,尤其涉及应用程序的崩溃信息处理方法及设备。

背景技术

[0002] 随着社会发展和软件技术的进步,智能终端上的应用程序也层出不穷,用户启动智能终端的应用程序(Application,APP)后,可能因为各种原因导致APP崩溃,对应用程序崩溃的相关信息进行收集,可以帮助技术人员定位出应用程序崩溃的具体原因。

[0003] 较为常见的引起APP线上崩溃的原因可能有:设备碎片化、带宽限制、网络变化、用户过多、代码错误、后端数据问题、内存溢出、未找到对应方法等多种类型,因为线上用户使用环境各不相同,所以要准确的判断造成崩溃的原因很困难,需要花费大量的人力和时间。

发明内容

[0004] 为克服相关技术中存在的问题,本申请提供一种应用程序的崩溃信息及设备,该应用程序的崩溃信息处理方法及设备能够快速确定应用程序发生崩溃的原因。

[0005] 本申请第一方面提供一种应用程序的崩溃信息处理方法,包括:

[0006] 判断所述应用程序上一次的关闭是否为异常关闭;

[0007] 若是,则将所述应用程序的崩溃信息发送至服务器,其中所述崩溃信息包括崩溃堆栈文件与已经保存的最近N条网络请求的具体信息;其中N为预先确定的正整数。

[0008] 本申请第二方面提供一种智能终端,包括:

[0009] 判断单元,用于判断所述应用程序上一次的关闭是否为异常关闭;

[0010] 发送单元,用于当所述判断单元的判断结果为是时,将所述应用程序的崩溃信息发送至服务器,其中所述崩溃信息包括崩溃堆栈文件与已经保存的最近N条网络请求的具体信息;其中N为预先确定的正整数。

[0011] 本申请第三方面提供一种电子设备,包括:

[0012] 处理器;以及

[0013] 存储器,其上存储有可执行代码,当所述可执行代码被所述处理器执行时,使所述处理器执行如上所述的方法。

[0014] 本申请第四方面提供一种非暂时性机器可读存储介质,其上存储有可执行代码,当所述可执行代码被电子设备的处理器执行时,使所述处理器执行如上所述的方法。

[0015] 本申请提供的技术方案可以包括以下有益效果:本申请实施例中,当应用程序发生崩溃错误时,终端在再次启动后会将崩溃堆栈信息和最近N条网络请求上传至服务器,服务器端的技术人员通过崩溃信息还原出应用发生崩溃时的网络环境,快速确定发生崩溃的原因。

[0016] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,并不能限制本申请。

附图说明

[0017] 通过结合附图对本申请示例性实施方式进行更详细的描述,本申请的上述以及其它目的、特征和优势将变得更加明显,其中,在本申请示例性实施方式中,相同的参考标号通常代表相同部件。

[0018] 图1是本申请实施例示出的应用程序的崩溃信息处理方法的第一流程示意图;

[0019] 图2是本申请实施例示出的应用程序的崩溃信息处理方法的第二流程示意图;

[0020] 图3是本申请实施例示出的应用程序的智能终端的结构示意图;

[0021] 图4是本申请实施例示出的电子设备的结构示意图。

具体实施方式

[0022] 下面将参照附图更详细地描述本申请的优选实施方式。虽然附图中显示了本申请的优选实施方式,然而应该理解,可以以各种形式实现本申请而不应被这里阐述的实施方式所限制。相反,提供这些实施方式是为了使本申请更加透彻和完整,并且能够将本申请的范围完整地传达给本领域的技术人员。

[0023] 在本申请使用的术语是仅仅出于描述特定实施例的目的,而非旨在限制本申请。在本申请和所附权利要求书中所使用的单数形式的“一种”、“所述”和“该”也旨在包括多数形式,除非上下文清楚地表示其他含义。还应当理解,本文中使用的术语“和/或”是指并包含一个或多个相关联的列出项目的任何或所有可能组合。

[0024] 应当理解,尽管在本申请可能采用术语“第一”、“第二”、“第三”等来描述各种信息,但这些信息不应限于这些术语。这些术语仅用来将同一类型的信息彼此区分开。例如,在不脱离本申请范围的情况下,第一信息也可以被称为第二信息,类似地,第二信息也可以被称为第一信息。由此,限定有“第一”、“第二”的特征可以明示或者隐含地包括一个或者更多个该特征。在本申请的描述中,“多个”的含义是两个或两个以上,除非另有明确具体的限定。

[0025] 相关技术中,查找程序崩溃原因主要依靠程序员的经验来判断造成线上崩溃的类型,通过收集的应用程序的崩溃堆栈信息与程序员的判断分析出很多崩溃都是因为前后端传递数据(以下统称接口)原因造成。但以目前技术手段,通过崩溃堆栈只能查看到线上应用程序用户有时候点击商品列表跳转到商品详情的时候会产生问题,无法完全还原用户当时使用的网络环境以及手机的运行情况,只能通过排除法,一步一步的依次排除后端数据问题、内存溢出……,这样一步一步排除,排查一个问题会花费大量的时间。假如花费大量时间后查出可能是因为商品的某个字段有问题,还得再次以排除法的方式查找,首先先去后端数据库查一下是否数据库存储的时候是否就有问题,如果存储没问题,再去排除是否在接口转换的时候是否有问题,因为无法精确定位具体的商品ID,排除工作是以亿位倍数的工作量在增长,因此相关技术中难以确定程序崩溃的真实原因,耗费的人力也极其巨大。针对上述问题,本申请实施例提供一种应用程序的崩溃信息处理方法,能够快速确定应用程序发生崩溃的具体原因。

[0026] 以下结合附图详细描述本申请实施例的技术方案。

[0027] 图1是本申请实施例示出的应用程序的崩溃信息处理方法的第一流程示意图。

[0028] 参见图1,本申请实施例提供一种应用程序的崩溃信息处理方法,包括:

[0029] S101、判断应用程序上一次的关闭是否为异常关闭,若是则执行S102,否则执行S103;

[0030] 本实施例中,应用程序可以是但不限于安装在手机、平板电脑、个人电脑等智能终端上的应用程序;

[0031] 举例来说,可以是在应用程序发生崩溃时、应用程序启动时,或者启动后的运行过程中执行S101中的判断;

[0032] S102、将上述应用程序的崩溃信息发送至服务器;

[0033] 如果应用程序上一次关闭为异常关闭,可能是因为应用程序崩溃导致,则将崩溃信息发送至服务器;

[0034] 其中上述崩溃信息包括崩溃堆栈文件与已经保存的最近N条网络请求的具体信息;

[0035] 需要说明的是,本申请中实施例中“网络请求”均指上述应用程序发出的网络请求;

[0036] 其中N为预先确定的正整数,例如N为1、2、3、5等数值;

[0037] 优选的,N可以为预先设定的固定值,也可以是动态变化的数值,例如可以依据应用程序在发生崩溃前的预定时间(例如1秒或3秒)内发出的网络请求数来确定,这样可以将发生崩溃前一定时间内的网络请求都上传给服务器;

[0038] S103、执行其它操作。

[0039] 如果应用程序上一次的关闭为正常关闭,例如按照用户的指令主动关闭,则无需上传崩溃信息。

[0040] 服务器在获取到崩溃信息后,服务器端的技术人员可以利用崩溃堆栈文件和最近N条网络请求的具体信息确定应用程序的崩溃原因是否与网络接口数据有关,如果有关,则再利用最近N条网络请求的具体信息确定具体是哪些网络接口数据出现了错误,导致应用程序发生崩溃。

[0041] 本实施例提供的应用程序的崩溃信息处理方法,如果应用程序发生崩溃,当应用程序再次启动后,应用程序会将崩溃堆栈文件与已经保存的最近N条网络请求的具体信息发送至服务器,服务器端的技术人员可以利用崩溃堆栈文件和最近N条网络请求的具体信息快速确定软件崩溃的具体原因。

[0042] 图2是本申请实施例示出的应用程序的崩溃信息处理方法的第二流程示意图。

[0043] 参见图2,本申请实施例提供的另一种应用程序的崩溃信息处理方法,包括:

[0044] S201、启动应用程序;

[0045] 本实施例中,应用程序可以是但不限于安装在手机、平板电脑、个人电脑等智能终端上的应用程序;

[0046] S202、判断上述应用程序是否为初次启动,如果是,执行S203,否则执行S204;

[0047] S203、在本地创建目标文件,上述目标文件用于保存网络请求的具体信息;

[0048] 如果应用程序为初次启动,在智能终端本地创建目标文件,该目标文件用于保存智能终端所发送网络请求的具体信息;

[0049] S204、判断上述应用程序的上一次关闭是否为异常关闭,若是则执行S205,否则执行S206;

- [0050] 应用程序发生崩溃后,可以自动重新启动,或者是按照用户输入的指令重新启动;
- [0051] S205、将上述应用程序的崩溃信息发送至服务器;
- [0052] 如果应用程序上一次关闭为异常关闭,可能是因为应用程序崩溃导致,则将崩溃信息发送至服务器;
- [0053] 其中上述崩溃信息包括崩溃堆栈文件与目标文件中已经保存的最近N条网络请求的具体信息;
- [0054] 其中N为预先确定的正整数,例如N为1、2、3、5等数值,优选的N可以为3;
- [0055] S206、执行其它操作。
- [0056] 如果应用程序上一次的关闭为正常关闭,例如按照用户的指令主动关闭,则无需上传崩溃信息。
- [0057] 优选的,该方法还可包括:当上述应用程序发送网络请求时,保存上述网络请求的具体信息;
- [0058] 具体的,可以是上述网络请求的具体信息写入上述目标文件中进行保存,可以将每一条网络请求都保存在目标文件中备用。
- [0059] 优选的,在上述目标文件中,各条网络请求的具体信息可以按照网络请求发送时间的先后顺序排列成队列,以便在发送崩溃信息时从队列前端提取前N条网络请求的具体信息,也即是上述最近N条网络请求的具体信息。
- [0060] 优选的,上述保存上述网络请求的具体信息包括:若上述目标文件中保存的网络请求的具体信息的数量达到第一预定阈值M(其中M大于N的正整数,例如M可以为10、20、50等数值),则删除上述队列中发送时间最早的至少一条(例如,可以是1条、3条、5条)网络请求的具体信息,并写入最新的网络请求的具体信息,这样使得目标文件中一直保存着最新的M条网络请求的具体信息;例如当M=20,若队列中已经存储了20条网络请求的具体信息,达到了上限,则删除队列中发送时间最早的3条网络请求的具体信息,以便写入最新的网络请求的具体信息。
- [0061] 或者,优选的,上述保存上述网络请求的具体信息包括:若上述目标文件中保存的网络请求的具体信息的数量达到第二预定阈值P(其中P大于N的正整数,例如P可以为10、20、50等数值),清空上述队列,并写入最新的网络请求的具体信息;网络请求的发送密度如果很高,可以在目标文件中保存的网络请求的具体信息的数量达到第二预定阈值P时,清空队列,以免造成存储延迟影响应用程序运行,例如当P=50,若队列中已经存储了50条网络请求的具体信息,达到了上限,则清空队列,以便写入最新的网络请求的具体信息。
- [0062] 举例来说,网络请求的具体信息可以包括但不限于:网络请求的发送信息与后端返回信息;
- [0063] 其中,网络请求的发送信息包括下列信息中的至少一种:网络请求的URL(Uniform Resource Locator,统一资源定位符)、上述网络请求的发送时间、上述网络请求所在页面;
- [0064] 其中,URL包含了用户信息,服务器后端可以根据URL去确定用户身份信息;网络请求的发送时间可以用于目标文件中网络请求的排序,以在发生崩溃时发送最新的N条网络请求的具体信息;崩溃堆栈文件中有应用程序的最后访问页面信息,网络请求所在页面的信息可以与崩溃堆栈文件中的最后的访问页面做比对,可以用于确定应用程序的崩溃跟哪条网络请求有关;

[0065] 网络请求的后端返回信息包括下列信息中的至少一种：上述网络请求是否成功、上述网络请求成功的具体信息、上述网络请求失败的具体信息；

[0066] 其中，服务器获取到网络请求的后端返回信息，可以利用网络请求成功或失败的具体信息确定哪些网络请求失败了，并确定是出现错误的信息中是哪个字段出现了错误。

[0067] 需要说明的是，服务器端获取到崩溃堆栈文件和最近N条网络请求的具体信息后，可以由技术人员对崩溃堆栈文件和最近N条网络请求的具体信息进行分析，确定出现错误的信息数据和字段是哪些，也可以是由服务器上部署有相关分析软件，自动地对崩溃堆栈文件和最近N条网络请求的具体信息进行分析，进而确定出现错误的信息数据和字段是哪些。

[0068] 优选的，该方法还可包括：服务器修复发生错误的网络接口数据；

[0069] 举例来说，服务器可以自动修复发生错误的网络接口数据，或者依操作服务器的用户（例如技术人员）输入的指令修复发生错误的网络接口数据。

[0070] 在确定了具体是哪些网络接口数据发生了错误，服务器可以修复发生错误的网络接口数据，应用程序再次启动后，就不会因同样的原因再次发生崩溃，服务器也无需发布新的应用程序版本或者修复软件包给智能终端，达到快速解决线上问题的目的。

[0071] 下面举例以说明本实施例提供的方案：用户在应用程序中通过一个列表点击可以跳转到一个商品的详情页面，在正常的情况下，商品详情页面都是可以打开的，但突然之间打开某一个商品时，应用程序发生了崩溃。在相关技术中，服务器只能获取到崩溃堆栈文件，然后要利用崩溃堆栈文件对崩溃原因进行逐一排查，非常耗时耗力。而应用本实施例提供的方案，可以利用崩溃堆栈文件和最近N条网络请求，迅速的找到产生崩溃的商品ID具体是哪个，如果该商品的数据有问题，通过网络请求的成功信息或失败信息就能确定商品数据的具体哪个字段有问题，则可以在后端修改此商品有问题的字段，这样其他用户或当前用户再次打开此商品时不再崩溃，从而从根本上解决应用程序产生崩溃的问题。

[0072] 与前述应用功能实现方法实施例相对应，本申请还提供了一种智能终端、电子设备及相应的实施例。

[0073] 图3是本申请实施例示出的智能终端的结构示意图。

[0074] 参见图3，该智能终端300包括：

[0075] 判断单元310，用于判断上述应用程序上一次的关闭是否为异常关闭；

[0076] 发送单元320，用于当上述判断单元310的判断结果为是时，将上述应用程序的崩溃信息发送至服务器。

[0077] 如果应用程序上一次关闭为异常关闭，可能是因为应用程序崩溃导致，则将崩溃信息发送至服务器；

[0078] 优选的，判断单元310可以是在应用程序启动时判断上述应用程序上一次的关闭是否为异常关闭，应用程序在崩溃时可以自行重新启动或者由用户重新启动，在重新启动后，可以将崩溃信息发送给服务器，以使得服务器端可以确定发生崩溃的具体原因。

[0079] 其中上述崩溃信息包括崩溃堆栈文件与已经保存的最近N条网络请求的具体信息；其中N为预先确定的正整数，例如N为1、2、3、5等数值。

[0080] 举例来说，智能终端300可以是手机、平板电脑、个人电脑（PC）等终端，判断单元310和发送单元320可以由软件系统实现的，也可以是由软硬件结合系统或者纯硬件系统

实现的,在此不做具体限制。

[0081] 优选的,N可以为预先设定的固定值(优选但不限于设置为3),也可以是动态变化的数值,例如可以依据应用程序在发生崩溃前的预定时间(例如1秒或3秒)内发出的网络请求数来确定,这样可以将发生崩溃前一定时间内的网络请求都上传给服务器;

[0082] 服务器在获取到崩溃信息后,可以利用崩溃堆栈文件确定应用程序的崩溃原因是否与网络接口数据有关,如果有关,则可利用终端上传的最近N条网络请求的具体信息确定具体是哪些网络接口数据出现了错误,导致应用程序发生崩溃。

[0083] 优选的,该智能终端300还可以包括:文件创建单元,用于在应用程序初次启动时,在智能终端本地创建目标文件,该目标文件用于保存智能终端所发送网络请求的具体信息。

[0084] 优选的,该智能终端300还可以包括:存储单元,用于当上述应用程序发送网络请求时,保存上述网络请求的具体信息;

[0085] 具体的,可以是将上述网络请求的具体信息写入上述目标文件中进行保存,可以将每一条网络请求都保存在目标文件中备用。

[0086] 优选的,在上述目标文件中,各条网络请求的具体信息可以按照网络请求发送时间的先后顺序进行排序,以便提取前N条。

[0087] 优选的,在上述目标文件中,各条网络请求的具体信息可以按照网络请求发送时间的先后顺序排列成队列,以便在发送崩溃信息时从队列前端提取前N条网络请求的具体信息,也即是上述最近N条网络请求的具体信息。

[0088] 优选的,上述存储单元可以具体用于:若上述目标文件中保存的网络请求的具体信息的数量达到第一预定阈值M(其中M大于N的正整数,例如M可以为10、20等数值),则删除上述队列中发送时间最早的至少一条(例如,可以是1条、3条、5条)网络请求的具体信息,并写入最新的网络请求的具体信息,这样使得目标文件中一直保存着最新的M条网络请求的具体信息。

[0089] 或者,优选的,上述存储单元可以具体用于:若上述目标文件中保存的网络请求的具体信息的数量达到第二预定阈值P(其中P大于N的正整数,例如P可以为10、20等数值),清空上述队列,并写入最新的网络请求的具体信息;网络请求的发送密度如果很高,可以在目标文件中保存的网络请求的具体信息的数量达到第二预定阈值P时,清空队列,以免造成存储延迟。

[0090] 举例来说,网络请求的具体信息可以包括但不限于:网络请求的发送信息与后端返回信息;

[0091] 其中,网络请求的发送信息包括下列信息中的至少一种:网络请求的URL(Uniform Resource Locator,统一资源定位符)、上述网络请求的发送时间、上述网络请求所在页面;

[0092] 网络请求的后端返回信息包括下列信息中的至少一种:上述网络请求是否成功、上述网络请求成功的具体信息、上述网络请求失败的具体信息。

[0093] 本申请还提供一种应用程序的崩溃信息处理系统,该系统包括如上所述的智能终端300和如上所述的服务器,因在前述实施例中对智能终端300和服务器进行了详细描述,在此不再赘述。

[0094] 关于上述实施例中的设备和系统,其中各个单元执行操作的具体方式已经在有关

该方法的实施例中进行了详细描述,此处将不再做详细阐述说明。

[0095] 图4是本申请实施例示出的电子设备的结构示意图。

[0096] 参见图4,电子设备400包括存储器410和处理器420。

[0097] 处理器420可以是中央处理单元(Central Processing Unit,CPU),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)、现场可编程门阵列(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。

[0098] 存储器410可以包括各种类型的存储单元,例如系统内存、只读存储器(ROM),和永久存储装置。其中,ROM可以存储处理器420或者计算机的其他模块需要的静态数据或者指令。永久存储装置可以是可读写的存储装置。永久存储装置可以是即使计算机断电后也不会失去存储的指令和数据而非易失性存储设备。在一些实施方式中,永久性存储装置采用大容量存储装置(例如磁或光盘、闪存)作为永久存储装置。另外一些实施方式中,永久性存储装置可以是可移除的存储设备(例如软盘、光驱)。系统内存可以是可读写存储设备或者易失性可读写存储设备,例如动态随机访问内存。系统内存可以存储一些或者所有处理器在运行时需要的指令和数据。此外,存储器410可以包括任意计算机可读存储媒介的组合,包括各种类型的半导体存储芯片(DRAM,SRAM,SDRAM,闪存,可编程只读存储器),磁盘和/或光盘也可以采用。在一些实施方式中,存储器410可以包括可读和/或写的可移除的存储设备,例如激光唱片(CD)、只读数字多功能光盘(例如DVD-ROM,双层DVD-ROM)、只读蓝光光盘、超密度光盘、闪存卡(例如SD卡、min SD卡、Micro-SD卡等等)、磁性软盘等等。计算机可读存储媒介不包含载波和通过无线或有线传输的瞬间电子信号。

[0099] 存储器410上存储有可执行代码,当可执行代码被处理器420处理时,可以使处理器420执行上文述及的方法中的部分或全部。

[0100] 举例来说,电子设备400可以是智能终端,或者服务器。

[0101] 上文中已经参考附图详细描述了本申请的方案。在上述实施例中,对各个实施例的描述都各有侧重,某个实施例中没有详细描述的部分,可以参见其他实施例的相关描述。本领域技术人员也应该知悉,说明书中所涉及的动作和模块并不一定是本申请所必须的。另外,可以理解,本申请实施例方法中的步骤可以根据实际需要进行顺序调整、合并和删减,本申请实施例装置中的模块可以根据实际需要进行合并、划分和删减。

[0102] 此外,根据本申请的方法还可以实现为一种计算机程序或计算机程序产品,该计算机程序或计算机程序产品包括用于执行本申请的上述方法中部分或全部步骤的计算机程序代码指令。

[0103] 或者,本申请还可以实施为一种非暂时性机器可读存储介质(或计算机可读存储介质、或机器可读存储介质),其上存储有可执行代码(或计算机程序、或计算机指令代码),当所述可执行代码(或计算机程序、或计算机指令代码)被电子设备(或电子设备、服务器等)的处理器执行时,使所述处理器执行根据本申请的上述方法的各个步骤的部分或全部。

[0104] 本领域技术人员还将明白的是,结合这里的申请所描述的各种示例性逻辑块、模块、电路和算法步骤可以被实现为电子硬件、计算机软件或两者的组合。

[0105] 附图中的流程图和框图显示了根据本申请的多个实施例的系统和方法的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段或代码的一部分,所述模块、程序段或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所标记的功能也可以以不同于附图中所标记的顺序发生。例如,两个连续的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意的,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合,可以用执行规定的功能或操作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0106] 以上已经描述了本申请的各实施例,上述说明是示例性的,并非穷尽性的,并且也不限于所披露的各实施例。在不偏离所说明的各实施例的范围和精神的情况下,对于本技术领域的普通技术人员来说许多修改和变更都是显而易见的。本文中所用术语的选择,旨在最好地解释各实施例的原理、实际应用或对市场中的技术的改进,或者使本技术领域的其它普通技术人员能理解本文披露的各实施例。

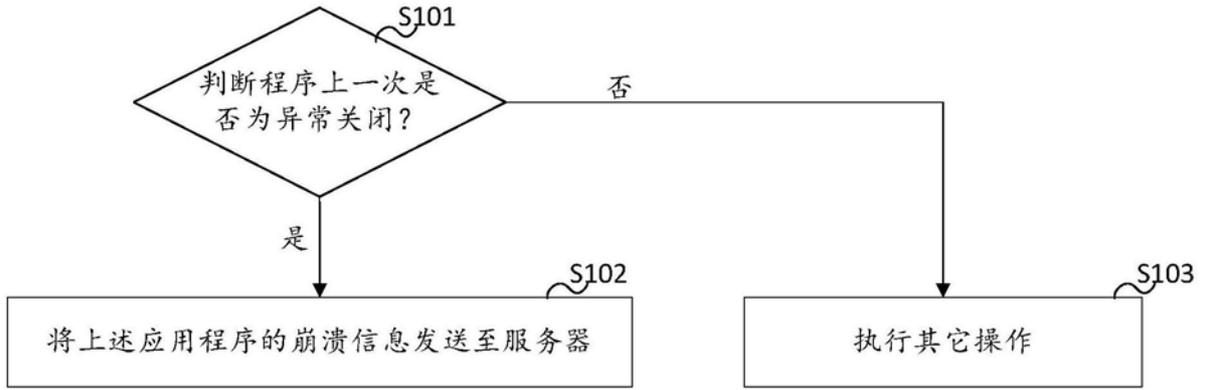


图1

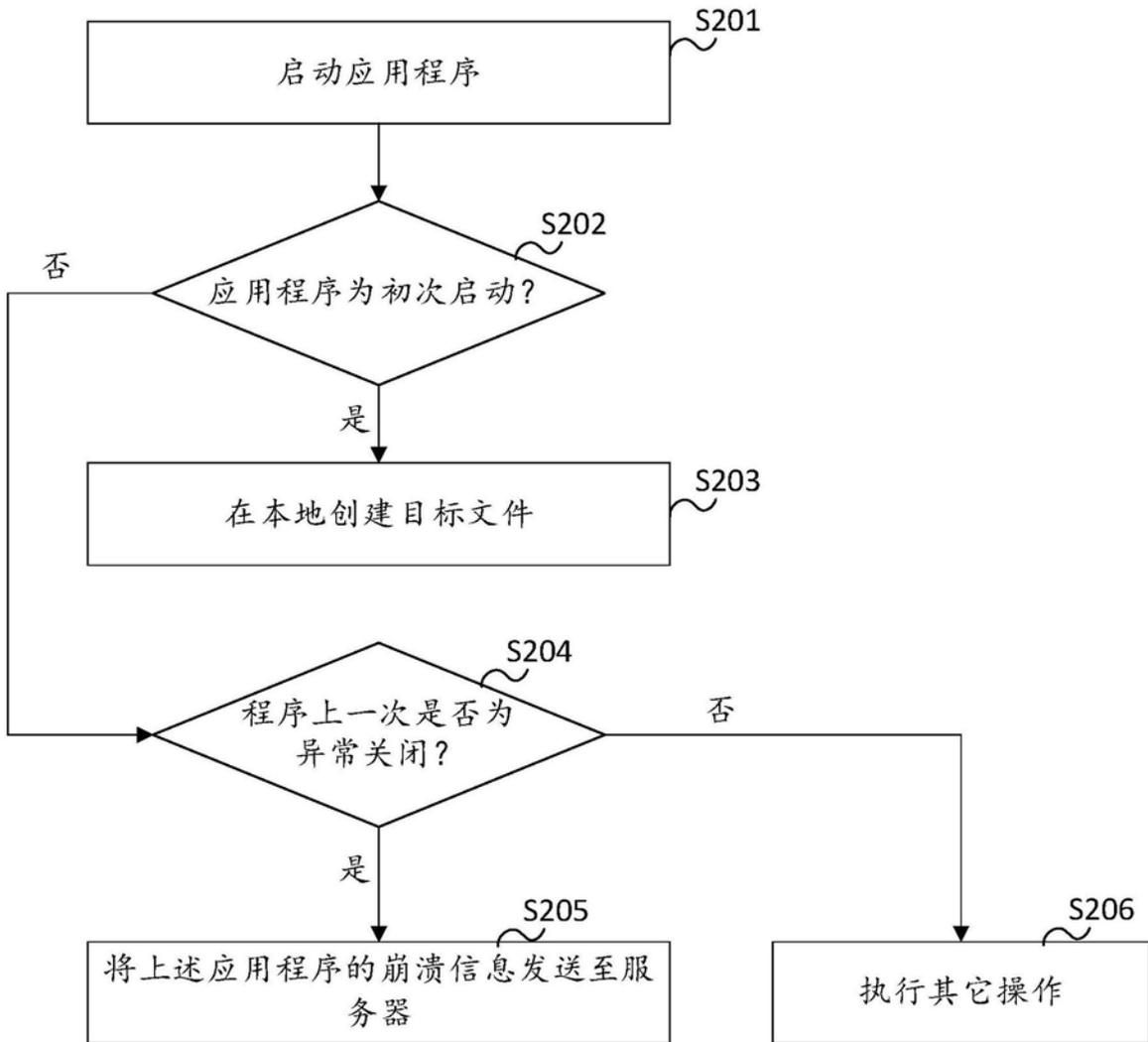


图2

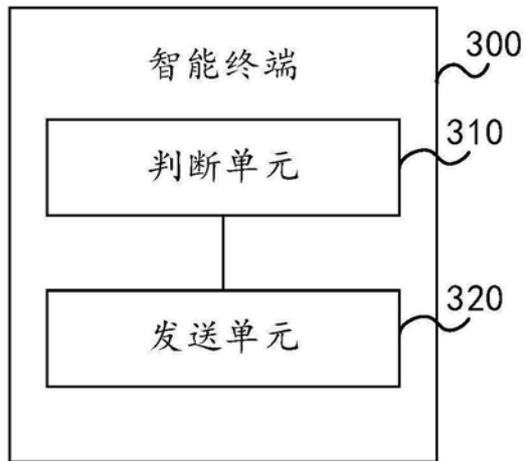


图3



图4