(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2018 年 12 月 20 日 (20.12.2018)

WIPO | PCT

(10) 国际公布号
WO 2018/227822 A1

(51) 国际专利分类号:
G06F 11/07 (2006.01)

(21) 国际申请号: PCT/CN2017/105018

(22) 国际申请日: 2017 年 9 月 30 日 (30.09.2017)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201710440288.8 2017年6月12日 (12.06.2017) CN

(71) 申请人: 平安科技（深圳）有限公司 (PING AN TECHNOLOGY (SHENZHEN) CO., LTD.) [CN/CN]; 中国广东省深圳市福田区八卦岭八卦三路平安大厦吴东勤, Guangdong 518000 (CN)。

(72) 发明人: 杜远 (DU, Yuan); 中国广东省深圳市福田区八卦岭八卦三路平安大厦吴东勤, Guangdong 518000 (CN)。 叶龙飞 (YE, Longfei); 中国广东省深圳市福田区八卦岭八卦三路平安大厦吴东勤, Guangdong 518000 (CN)。

(74) 代理人: 深圳市沃德知识产权代理事务所（普通合伙） (SHENZHEN WORLD INTELLECTUAL PROPERTY AGENCY (GENERAL PARTNERSHIP)); 中国广东省深圳市福田区园岭街道八卦四路10号中浩大厦1528-1530室于志光, Guangdong 518000 (CN)。

(81) 指定国（除另有指明，要求每一种可提供的国家保护）: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,

(54) **Title:** ACQUISITION METHOD AND APPARATUS FOR ABNORMAL STACK INFORMATION, AND COMPUTER READABLE STORAGE MEDIUM

(54) **发明名称**: 异常堆栈信息获取方法、装置及计算机可读存储介质

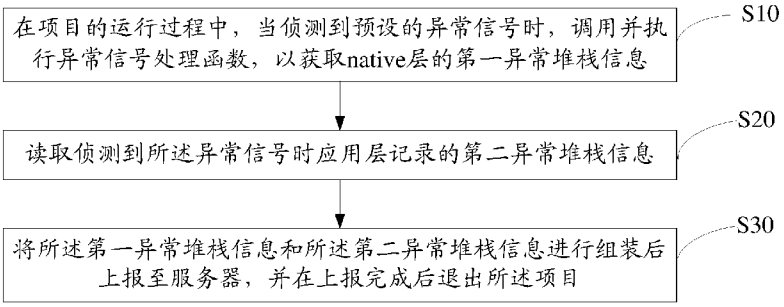

图 1

S10 Call and execute an abnormal signal processing function when a preset abnormal signal is detected during the running of a project so as to acquire first abnormal stack information of a native level

S20 Read second abnormal stack information recorded by an application level when said abnormal signal is detected

S30 Package said first abnormal stack information and said second abnormal stack information and report the same to a server, and exit said project once the reporting is complete

(57) **Abstract:** An acquisition method and apparatus for abnormal stack information, and a computer readable storage medium, wherein said method comprises: calling and executing an abnormal signal processing function when a preset abnormal signal is detected during the running of a project so as to acquire first abnormal stack information of a native level (S10); reading second abnormal stack information recorded by an application level when said abnormal signal is detected (S20); and packaging said first abnormal stack information and said second abnormal stack information and reporting the same to a server, and exiting said project once the reporting is complete (S30). The described method solves the technical problem in existing technology whereby it is difficult to acquire and upload abnormal stack information from an application program in a timely manner.

WO 2018/227822 A1

[见续页]

SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国(除另有指明，要求每一种可提供的地区保护)：ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

(57) 摘要：一种异常堆栈信息获取方法、装置以及计算机可读存储介质，该方法包括：在项目的运行过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取native层的第一异常堆栈信息(S10)；读取侦测到所述异常信号时应用层记录的第二异常堆栈信息(S20)；将所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目(S30)。该方法解决了现有技术中难以及时地获取应用程序中的异常堆栈信息并上传的技术问题。

异常堆栈信息获取方法、装置及计算机可读存储介质

优先权申明

本申请基于巴黎公约申明享有2017年06月12日递交的申请号为CN 201710440288.8、名称为“异常堆栈信息获取方法、装置及计算机可读存储介质”的中国专利申请的优先权，该中国专利申请的整体内容以参考的方式结合在本申请中。

技术领域

本发明涉及计算机技术领域，尤其涉及一种异常堆栈信息获取方法、装置及计算机可读存储介质。

背景技术

目前，如果 APP 在程序设计中存在异常，则用户在使用 APP 的过程中，会出现因 APP 的运行异常而发生崩溃闪退的现象，导致 APP 不能正常使用。因此，对于 APP 开发者来说，需要获取到应用程序在发生异常时产生的异常堆栈信息，及时地对应用程序中产生的异常问题的代码进行定位并分析。

而现有的获取异常堆栈信息的方式一般是，在应用程序发生异常并崩溃后，程序再次启动时，会从本地获取异常堆栈信息并上传。但是这种处理方式存在一定的缺陷，如果应用程发生异常并崩溃后，用户没有再次启动应用，则无法上传。

发明内容

本发明提供一种异常堆栈信息获取方法、装置及计算机可读存储介质，其主要目的在于解决现有技术中难以及时地获取应用程序中的异常堆栈信息并上传的技术问题。

为实现上述目的，本发明提供一种异常堆栈信息获取方法，该方法包括：

在项目的运行过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取 native 层的第一异常堆栈信息；

读取侦测到所述异常信号时应用层记录的第二异常堆栈信息；

将所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目。

可选地，所述在项目的运行过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取 native 层的第一异常堆栈信息的步骤包括：

在所述项目的运行过程中，当侦测到预设的异常信号时，确定运行所述项目的操作系统的版本信息；

调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息，并根据所述地址信息生成第一异常堆栈信息。

可选地，所述调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息，并根据所述地址信息生成第一异常堆栈信息的步骤包括：

调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息；

根据所述地址信息获取异常堆栈数据，获取侦测到的所述异常信号的信号类型；

将所述信号类型与所述异常堆栈数据组装后生成所述第一异常堆栈信息。

可选地，所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目的步骤包括：

读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息；

将所述第三异常堆栈信息与所述第二异常堆栈信息进行组装后上报至所述服务器，并在上报完成后退出所述项目。

此外，为实现上述目的，本发明还提供一种异常堆栈信息获取装置，该装置包括：存储器、处理器及存储在所述存储器上并可在所述处理器上运行的异常堆栈信息获取程序，所述异常堆栈信息获取程序被所述处理器执行时实现如下步骤：

在项目的运行过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取 native 层的第一异常堆栈信息；

读取侦测到所述异常信号时应用层记录的第二异常堆栈信息；

将所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目。

可选地，所述在项目的运行过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取 native 层的第一异常堆栈信息的步骤包括：

在所述项目的运行过程中，当侦测到预设的异常信号时，确定运行所述项目的操作系统的版本信息；

调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息，并根据所述地址信息生成第一异常堆栈信息。

可选地，所述调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息，并根据所述地址信息生成第一异常堆栈信息的步骤包括：

调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息；

根据所述地址信息获取异常堆栈数据，获取侦测到的所述异常信号的信号类型；

将所述信号类型与所述异常堆栈数据组装后生成所述第一异常堆栈信息。

可选地，所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目的步骤包括：

读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息；

将所述第三异常堆栈信息与所述第二异常堆栈信息进行组装后上报至所述服务器，并在上报完成后退出所述项目。

可选地，所述读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息的步骤包括：

启动信息还原脚本，基于所述信息还原脚本，将所述第一异常堆栈信息中的函数名依次与所述符号表中的函数名进行一一匹配，获取到所述第一异常堆栈信息中的函数名对应的文件名和行号；

将获取的文件名和行号添加至原始的第一异常堆栈信息，生成所述服务器可读的第三异常堆栈信息。

此外，为实现上述目的，本发明还提供一种计算机可读存储介质，所述计算机可读存储介质上存储有异常堆栈信息获取程序，所述异常堆栈信息获取程序被处理器执行时实现如下步骤：

在项目的运行过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取 native 层的第一异常堆栈信息；

读取侦测到所述异常信号时应用层记录的第二异常堆栈信息；

将所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目。

本发明提出的异常堆栈信息获取方法、装置及计算机可读存储介质，在项目的运行过程中侦测到预设的异常信号时，调用并执行异常信号处理函数获取 native 层的第一异常堆栈信息，并且读取侦测到异常信号时应用层记录的第二异常堆栈信息，将上述 native 层和应用层的异常堆栈信息进行组装后上报至服务器，并且在上报完成后退出该项目，避免在检测到异常后，直接强制关闭项目，并且该发明能够通过预先定义的异常信号处理函数及时获取 native 层的第一异常堆栈信息在内的完整异常堆栈信息上传至服务器，能够全面的监控异常的发生，实时上报异常堆栈信息，对于开发人员来说，可以及时发现应用的异常，以便于快速进行异常定位和解决问题。

附图说明

图1为本发明异常堆栈信息获取方法第一实施例的流程图；

图2为本发明异常堆栈信息获取方法第二实施例中步骤S30的细化流程示意图；

图3为本发明异常堆栈信息获取装置较佳实施例的示意图；

图4为本发明异常堆栈信息获取装置中的异常堆栈信息获取程序的功能模块示意图。

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

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。

本发明提供一种异常堆栈信息获取方法。参照图 1 所示，为本发明异常堆栈信息获取方法第一实施例的流程图。该方法可以由一个装置执行，该装置可以由软件和/或硬件实现。

在本实施例中，该异常堆栈信息获取方法包括：

步骤 S10，在项目的运行过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取 native 层的第一异常堆栈信息。

本发明实施例中，项目是由代码构成的应用程序。终端可以从服务器下载项目，并安装运行。项目在终端上的运行过程中，对项目的运行情况进行实时检测。预先设置各种预设类型的异常信号，例如，</br>

SIGABRT：由调用 abort 函数产生，进程非正常退出</br>

SIGILL：非法指令异常</br>

SIGBUS：某种特定的硬件异常，通常有内存访问引起</br>

SIGFPE：数学相关异常，如被除 0、浮点移除等</br>

SIGSEGV：非法内存访问</br>

在项目的运行过程中，当项目发生相关异常时，会触发对应的异常信号，来通知进程发生了异步事件。当终端侦测到的异常信号属于预设的异常信号时，调用并执行预设的异常信号处理函数，以获取 native 层的第一异常堆栈信息。异常信号处理函数需要预先配置，以供调用。

作为一种实施方式，异常信号处理函数可以通过以下代码实现：

<pre><code>

```
/* Signals to be caught. */
#define SIG_CATCH_COUNT 5
static const int native_sig_catch[SIG_CATCH_COUNT]= {SIGABRT, SIGILL, SIGBUS, SIGFPE, SIGSEGV};
/*
 * 注册异常信号处理函数
 */
int sigaction_handler_init(void) {
    struct sigaction sa_catch;
    sa_catch.sa_sigaction = skn_signal_handler;
    sa_catch.sa_flags = SA_SIGINFO;
    size_t i = 0;
    for (i = 0; native_sig_catch[i] != 0; i++) {
        const int sig = native_sig_catch[i];
        const struct sigaction *const action = &sa_catch;
        if (sigaction(sig, action, NULL) != 0) {
            return -1;
        }
    }
    return 0;
}
```

</code></pre>

其中，函数_skn_signal_handler_就是检测到异常信号后执行的函数，异常堆栈信息的获取就是在这个函数中进行的。

由于终端使用的操作系统的版本不同，配置的异常信号处理函数也不相同。因此，步骤 S10 可以包括以下细化步骤：

在所述项目的运行过程中，当侦测到预设的异常信号时，确定运行所述项目的操作系统的版本信息；调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息，并根据所述地址信息生成第一异常堆栈信息。

具体地，若操作系统的版本低于预设版本，则调用第一预设函数获取发生异常的地址信息，通过 dladdr 函数获取与地址信息对应的函数名以及 so 库信息；若操作系统的版本高于或等于预设版本，则通过 dlopen 函数从预设路径中调用第二预设函数获取发生异常的地址信息，通过 dladdr 函数获取与地址信息对应的函数名以及 so 库信息。其中，第一预设函数为预先配置的异常信号处理函数，第二预设函数为操作系统定义的函数，不需要用户预先配置，只需在侦测到的异常信号时，从特定的路径中调用即可。在本实施例中，异常堆栈信息主要包括发生异常的函数名以及 so 库信息。

以下以 Android 系统为例，预设版本为 Android 5.0，由于 Android 5.0 开始 Android 采用 ART 虚拟机，所以异常信息的获取在 Android5.0 以上和 Android5.0 以下需要两种方案，Android 5.0 以上使用 unwind 方法，Android 5.0 以下使用 libcorkscrew.so 方法。具体的，关于 unwind 方法的使用，可以在代码中引入 unwind.h 头文件，通过 _Unwind_Backtrace 函数获取到产生异常的地址信息后，再通过 dladdr 函数获取到地址信息对应的函数名和 so 库信息。关于 libcorkscrew.so 方法的使用，上述预设路径可以是 Android4.x 版本的/system/lib/路径，通过 dlopen 函数打开对应的库，从而调用 libcorkscrew.so 方法，基于该方法查询到产生异常的地址信息，在基于该地址信息查找对应的函数名和 so 库信息。

优选地，作为一种实施方式，为了使开发人员能够了解到异常信号的具体信息，第一异常堆栈信息还包括侦测到的异常信号的信号类型。

调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息，并根据所述地址信息生成第一异常堆栈信息的步骤包括：调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息；根据所述地址信息获取异常堆栈数据，获取侦测到的所述异常信号的信号类型；将所述信号类型与所述异常

堆栈数据组装后生成所述第一异常堆栈信息。

根据发生异常的地址信息获取到对应的有异常堆栈数据后，获取侦测到异常信号时记录的该异常信号所述的信号类型，将该信号类型与上述异常堆栈数据组装以生成第一异常堆栈信息，异常堆栈数据包括上述查找到的异常的函数名和 so 库信息。其中，第一异常堆栈信息可以在 native 层获取并保存在本地后，由 native 层向应用层发送通知信息，通知应用层读取，对于 Android 系统老说，应用层即 Java 层。在 native 层组装信号类型与异常堆栈数据时，使用预设的分隔符进行分隔，以便于应用层区分。

步骤 S20，读取侦测到所述异常信号时应用层记录的第二异常堆栈信息。

步骤 S30，将所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目。

一般情况下，终端在侦测到异常信号时，其应用层会对其异常堆栈信息进行记录。一般根据系统的设置会存储在固定的路径中，因此，可以在检测到异常信号时，直接从设定的路径中读取存储的异常信息，作为上述第二异常堆栈信息。

将获取到的 native 层的第一异常堆栈信息和读取到的记录的应用层的第二异常堆栈信息进行封装处理，并上报到服务器，以供开发人员及时对应用程序中产生异常的代码进行定位并非分析，进而及时地对应用程序进行维护并更新。在上报完成后，退出当前的项目。

现有技术中获取 native 层异常堆栈的方式，一般代码量大，如 Google BreakPad，或者需要系统 root 权限，如 tombstone，才能够读取相关堆栈信息，且过程过于繁琐。

本实施例提出的异常堆栈信息获取方法，在项目的运行过程中侦测到预设的异常信号时，通过预先定义的函数获取到 native 层的异常堆栈信息，代码量小，且不需要 root 权限并且读取侦测到异常信号时应用层记录的第二异常堆栈信息，将两个异常堆栈信息封装后上传到服务器，再退出应用，避免在检测到异常后，直接强制关闭应用。因此能够及时的获取到包括 native 层在内的完整异常堆栈信息上传至服务器，能够全面的监控异常的发生，实时上报异常堆栈信息，对于开发人员来说，可以及时发现应用的异常，以便于快速定位和解决问题。

基于第一实施例提出本发明异常堆栈信息获取方法的第二实施例。参照图 2 所示，在本实施例中，步骤 S30 包括以下细化步骤：

步骤 S31，读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行

还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息；

步骤 S32，将所述第三异常堆栈信息与所述第二异常堆栈信息进行组装后上报至所述服务器，并在上报完成后退出所述项目。

一般情况下，通过上述方式获取到的第一异常堆栈信息，对于开发者来说，可读性比较低，从第一异常信息中只能简单的看出其函数名，无法直接了解到异常所在的文件名以及所在的行号。为了提高其可读性，在获取到第一异常堆栈信息后，通过该应用匹配的符号表来对第一异常堆栈信息进行还原处理，以得到包含有函数名、该函数名对应的文件名以及行号的第三异常堆栈信息。其中，符号表是一种用于语言翻译器的数据结构，在符号表中，程序源代码中每个标识符都和它的声明或者使用信息绑定在一起，在该实施例中，在项目的符号表中，每一个函数名都有其对应的文件名及所在的行号。每一个版本的项目对应一个符号表，若项目升级，则需要对其符号表进行更新。

优选地，在其他实施例中，为了提高项目信息的安全性，可以将符号表存放在服务器，服务器接收到终端上报的异常堆栈信息后，根据存储的符号表对第一异常堆栈信息进行还原处理。

本实施例中，预先配置有信息还原脚本，获取到第一异常堆栈信息后，启动信息还原脚本，基于所述信息还原脚本，将所述第一异常堆栈信息中的函数名依次与所述符号表中的函数名进行一一匹配，获取到所述第一异常堆栈信息中的函数名对应的文件名和行号；将获取的文件名和行号添加至原始的第一异常堆栈信息，生成所述服务器可读的第三异常堆栈信息。

信息还原脚本可以是：

<code><pre>

./SknSymbol.sh crash.txt symbol.txt >result.txt

</code></pre>

其中，crash.txt 为未还原的异常信息，即第一异常堆栈信息；symbol.txt 为符号表；result.txt 为还原后的异常信息，即第三异常堆栈信息。

例如，若获取到的异常信息为：

<code><pre>

_Z2aav /data/app/com.pingan.pad_demo-2/lib/arm/libcrash.so

</code></pre>

经过上述信息还原脚本还原后的异常信息为：

<code><pre>

Z2aav

/Users/duyuan797/Workspace/sky-eye/sky-eye/src/main/cpp/./src/skncrash.cp

p:72 /data/app/com.pingan.pad_demo-2/lib/arm/libcrash.so

</code></pre>

本实施例提出的方法，通过对获取到的 native 层的第一异常堆栈信息进行还原处理，以提高服务器对其的可读性，便于开发者获取到更加完整的异常信息。

参照图 3 所示，为本发明异常堆栈信息获取装置较佳实施例的示意图。

在本实施例中，异常堆栈信息获取装置可以是智能手机、平板电脑、电子书阅读器、MP3（Moving Picture Experts Group Audio Layer III，动态影像专家压缩标准音频层面 3）播放器、MP4（Moving Picture Experts Group Audio Layer IV，动态影像专家压缩标准音频层面 4）播放器、便携计算机等具有显示功能的可移动式终端设备。

该异常堆栈信息装置包括存储器 11、处理器 12，通信总线 13，以及网络接口 14。其中，通信总线 13 用于实现这些组件之间的连接通信。网络接口 14 可选的可以包括标准的有线接口、无线接口（如 WI-FI 接口）。存储器 11 可以是高速 RAM 存储器，也可以是稳定的存储器（non-volatile memory），例如磁盘存储器。

存储器 11 在一些实施例中可以是异常堆栈信息获取装置的内部存储单元，例如该异常堆栈信息获取装置的硬盘。存储器 11 在另一些实施例中也可以是异常堆栈信息获取装置的外部存储设备，例如异常堆栈信息获取装置上配备的插接式硬盘，智能存储卡（Smart Media Card, SMC），安全数字（Secure Digital, SD）卡，闪存卡（Flash Card）等。进一步地，存储器 11 还可以既包括异常堆栈信息获取装置的内部存储单元也包括外部存储设备。存储器 11 不仅可以用于存储安装于异常堆栈信息获取装置的应用软件及各类数据，例如异常堆栈信息获取程序的代码等，还可以用于暂时地存储已经输出或者将要输出的数据。

处理器 12 在一些实施例中可以是一中央处理器（Central Processing Unit, CPU），微处理器或其他数据处理芯片，用于运行存储器 11 中存储的程序代码或处理数据，例如执行异常堆栈信息获取程序等。

图 3 仅示出了具有组件 11-14 以及异常堆栈信息获取程序的异常堆栈信息获取装置，但是应理解的是，并不要求实施所有示出的组件，可以替代的实施更多或者更少的组件。

可选地，该装置还可以包括用户接口，用户接口可以包括显示器（Display）、输入单

元比如键盘（Keyboard），可选的用户接口还可以包括标准的有线接口、无线接口。在一些实施例中，显示器可以是 LED 显示器、液晶显示器、触控式液晶显示器以及 OLED（Organic Light-Emitting Diode，有机发光二极管）触摸器等。显示器用于显示在异常堆栈信息获取装置中处理的信息以及用于显示可视化的用户界面。

可选地，该装置还可以包括摄像头、RF（Radio Frequency，射频）电路，传感器、音频电路、WiFi 模块等等。其中，传感器比如光传感器、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，若该装置为移动终端，环境光传感器可根据环境光线的明暗来调节显示屏的亮度，接近传感器可在移动终端移动到耳边时，关闭显示屏和/或背光。作为运动传感器的一种，重力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，可用于识别移动终端姿态的应用（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；当然，移动终端还可配置陀螺仪、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。

在图 3 所示的装置实施例中，作为一种计算机存储介质的存储器 11 中可以包括操作系统、以及异常堆栈信息获取程序；网络接口 14 主要用于连接服务器，与服务器进行数据通信；处理器 12 执行存储器 11 中存储的异常堆栈信息获取程序时实现如下步骤：

在项目的运行过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取 native 层的第一异常堆栈信息；

读取侦测到所述异常信号时应用层记录的第二异常堆栈信息；

将所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目。

本发明实施例中，项目是由代码构成的应用程序。终端可以从服务器下载项目，并安装运行。项目在终端上的运行过程中，对项目的运行情况进行实时检测。预先设置各种预设类型的异常信号，例如，</br>

SIGABRT：由调用 abort 函数产生，进程非正常退出</br>

SIGILL：非法指令异常</br>

SIGBUS：某种特定的硬件异常，通常有内存访问引起</br>

SIGFPE：数学相关异常，如被除 0、浮点移除等</br>

SIGSEGV：非法内存访问</br>

在项目的运行过程中，当项目发生相关异常时，会触发对应的异常信号，来通知进程

发生了异步事件。当终端侦测到的异常信号属于预设的异常信号时，调用并执行预设的异常信号处理函数，以获取 native 层的第一异常堆栈信息。异常信号处理函数需要预先配置，以供调用。

作为一种实施方式，异常信号处理函数可以通过以下代码实现：

```
<pre><code>
/* Signals to be caught. */
#define SIG_CATCH_COUNT 5
static const int native_sig_catch[SIG_CATCH_COUNT]= {SIGABRT, SIGILL, SIGBUS, SIGFPE, SIGSEGV};
/*
 * 注册异常信号处理函数
 */
int sigaction_handler_init(void) {
    struct sigaction sa_catch;
    sa_catch.sa_sigaction = skn_signal_handler;
    sa_catch.sa_flags = SA_SIGINFO;
    size_t i = 0;
    for (i = 0; native_sig_catch[i] != 0; i++) {
        const int sig = native_sig_catch[i];
        const struct sigaction *const action = &sa_catch;
        if (sigaction(sig, action, NULL) != 0) {
            return -1;
        }
    }
    return 0;
}
</code></pre>
```

其中，函数_skn_signal_handler_就是检测到异常信号后执行的函数，异常堆栈信息的获取就是在这个函数中进行的。

由于终端使用的操作系统的版本不同，配置的异常信号处理函数也不相同。因此，异

常堆栈信息获取程序被处理器 12 执行时还实现如下步骤：

在所述项目的运行过程中，当侦测到预设的异常信号时，确定运行所述项目的操作系统的版本信息；调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息，并根据所述地址信息生成第一异常堆栈信息。

具体地，异常堆栈信息获取程序被处理器 12 执行时还实现如下步骤：

若操作系统的版本低于预设版本，则调用第一预设函数获取发生异常的地址信息，通过 dladdr 函数获取与地址信息对应的函数名以及 so 库信息；若操作系统的版本高于或等于预设版本，则通过 dlopen 函数从预设路径中调用第二预设函数获取发生异常的地址信息，通过 dladdr 函数获取与地址信息对应的函数名以及 so 库信息。

其中，第一预设函数为预先配置的异常信号处理函数，第二预设函数为操作系统定义的函数，不需要用户预先配置，只需在侦测到的异常信号时，从特定的路径中调用即可。在本实施例中，异常堆栈信息主要包括发生异常的函数名以及 so 库信息。

以下以 Android 系统为例，预设版本为 Android 5.0，由于 Android 5.0 开始 Android 采用 ART 虚拟机，所以异常信息的获取在 Android5.0 以上和 Android5.0 以下需要两种方案，Android 5.0 以上使用 unwind 方法，Android 5.0 以下使用 libcorkscrew.so 方法。具体的，关于 unwind 方法的使用，可以在代码中引入 unwind.h 头文件，通过 _Unwind_Backtrace 函数获取到产生异常的地址信息后，再通过 dladdr 函数获取到地址信息对应的函数名和 so 库信息。关于 libcorkscrew.so 方法的使用，上述预设路径可以是 Android4.x 版本的/system/lib/路径，通过 dlopen 函数打开对应的库，从而调用 libcorkscrew.so 方法，基于该方法查询到产生异常的地址信息，在基于该地址信息查找对应的函数名和 so 库信息。

优选地，作为一种实施方式，为了使开发人员能够了解到异常信号的具体信息，第一异常堆栈信息还包括侦测到的异常信号的信号类型。

异常堆栈信息获取程序被处理器 12 执行时还实现如下步骤：

调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息，并根据所述地址信息生成第一异常堆栈信息的步骤包括：调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息；根据所述地址信息获取异常堆栈数据，获取侦测到的所述异常信号的信号类型；将所述信号类型与所述异常堆栈数据组装后生成所述第一异常堆栈信息。

根据发生异常的地址信息获取到对应的有异常堆栈数据后，获取侦测到异常信号时记

录的该异常信号所述的信号类型，将该信号类型与上述异常堆栈数据组装以生成第一异常堆栈信息，异常堆栈数据包括上述查找到的异常的函数名和 so 库信息。其中，第一异常堆栈信息可以在 native 层获取并保存在本地后，由 native 层向应用层发送通知信息，通知应用层读取，对于 Android 系统老说，应用层即 Java 层。在 native 层组装信号类型与异常堆栈数据时，使用预设的分隔符进行分隔，以便于应用层区分。

一般情况下，终端在侦测到异常信号时，其应用层会对其异常堆栈信息进行记录。一般根据系统的设置会存储在固定的路径中，因此，可以在检测到异常信号时，直接从设定的路径中读取存储的异常信息，作为上述第二异常堆栈信息。

将获取到的 native 层的第一异常堆栈信息和读取到的记录的应用层的第二异常堆栈信息进行封装处理，并上报到服务器，以供开发人员及时对应用程序中产生异常的代码进行定位并非分析，进而及时地对应用程序进行维护并更新。在上报完成后，退出当前的项目。

现有技术中获取 native 层异常堆栈的方式，一般代码量大，如 Google BreakPad，或者需要系统 root 权限，如 tombstone，才能够读取相关堆栈信息，且过程过于繁琐。

本实施例提出的异常堆栈信息获取装置，在项目的运行过程中侦测到预设的异常信号时，通过预先定义的函数获取到 native 层的异常堆栈信息，代码量小，且不需要 root 权限并且读取侦测到异常信号时应用层记录的第二异常堆栈信息，将两个异常堆栈信息封装后上传到服务器，再退出应用，避免在检测到异常后，直接强制关闭应用。因此能够及时的获取到包括 native 层在内的完整异常堆栈信息上传至服务器，能够全面的监控异常的发生，实时上报异常堆栈信息，对于开发人员来说，可以及时发现应用的异常，以便于快速定位和解决问题。

基于第一实施例提出本发明异常堆栈信息获取装置的第二实施例。在本实施例中，异常堆栈信息获取程序被处理器 12 执行时还实现如下步骤：

读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息；

将所述第三异常堆栈信息与所述第二异常堆栈信息进行组装后上报至所述服务器，并在上报完成后退出所述项目。

一般情况下，通过上述方式获取到的第一异常堆栈信息，对于开发者来说，可读性比较低，从第一异常信息中只能简单的看出其函数名，无法直接了解到异常所在的文件名以及所在的行号。为了提高其可读性，在获取到第一异常堆栈信息后，通过该应用匹配的符

号表来对第一异常堆栈信息进行还原处理，以得到包含有函数名、该函数名对应的文件名以及行号的第三异常堆栈信息。其中，符号表是一种用于语言翻译器的数据结构，在符号表中，程序源代码中每个标识符都和它的声明或者使用信息绑定在一起，在该实施例中，在项目的符号表中，每一个函数名都有其对应的文件名及所在的行号。每一个版本的项目对应一个符号表，若项目升级，则需要对其符号表进行更新。

优选地，在其他实施例中，为了提高项目信息的安全性，可以将符号表存放在服务器，服务器接收到终端上报的异常堆栈信息后，根据存储的符号表对第一异常堆栈信息进行还原处理。

本实施例中，预先配置有信息还原脚本，获取到第一异常堆栈信息后，处理器 12 执行异常堆栈信息获取程序，实现如下步骤：

启动信息还原脚本，基于所述信息还原脚本，将所述第一异常堆栈信息中的函数名依次与所述符号表中的函数名进行一一匹配，获取到所述第一异常堆栈信息中的函数名对应的文件名和行号；将获取的文件名和行号添加至原始的第一异常堆栈信息，生成所述服务器可读的第三异常堆栈信息。

信息还原脚本可以是：

<code><pre>

./SknSymbol.sh crash.txt symbol.txt >result.txt

</code></pre>

其中，crash.txt 为未还原的异常信息，即第一异常堆栈信息；symbol.txt 为符号表；result.txt 为还原后的异常信息，即第三异常堆栈信息。

例如，若获取到的异常信息为：

<code><pre>

_Z2aav /data/app/com.pingan.pad_demo-2/lib/arm/libcrash.so

</code></pre>

经过上述信息还原脚本还原后的异常信息为：

<code><pre>

Z2aav

/Users/duyuan797/Workspace/sky-eye/sky-eye/src/main/cpp/./src/skncrash.cpp:72 /data/app/com.pingan.pad_demo-2/lib/arm/libcrash.so

</code></pre>

本实施例提出的装置，通过异常堆栈信息获取程序获取到的 native 层的第一异常堆栈信息进行还原处理，以提高服务器对其的可读性，便于开发者获取到更加完整的异常信息。

可选地，在其他的实施例中，异常堆栈信息获取程序还可以被分割为一个或者多个模块，一个或者多个模块被存储于存储器 11 中，并由一个或多个处理器（本实施例为处理器 12）所执行，以完成本发明。本发明所称的模块是指能够完成特定功能的一系列计算机程序指令段，用于描述异常堆栈信息获取程序在异常堆栈信息获取装置中的执行过程。

例如，在一实施例中，异常堆栈信息获取程序可以被分割为获取模块 10 和上报模块 20，参照图 4 所示，为本发明异常堆栈信息获取装置中的异常堆栈信息获取程序的功能模块示意图，其中：

获取模块 10，用于在项目的运行过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取 native 层的第一异常堆栈信息，以及读取侦测到所述异常信号时应用层记录的第二异常堆栈信息；

上报模块 20，用于将所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目。

此外，本发明实施例还提出一种计算机可读存储介质，所述计算机可读存储介质上存储有异常堆栈信息获取程序，所述异常堆栈信息获取程序被处理器执行时实现如下操作：

在项目的运行过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取 native 层的第一异常堆栈信息；

读取侦测到所述异常信号时应用层记录的第二异常堆栈信息；

将所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目。

进一步地，所述异常堆栈信息获取程序被处理器执行时还实现如下操作：

在所述项目的运行过程中，当侦测到预设的异常信号时，确定运行所述项目的操作系统的版本信息；

调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息，并根据所述地址信息生成第一异常堆栈信息。

进一步地，所述异常堆栈信息获取程序被处理器执行时还实现如下操作：

调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地

址信息；

根据所述地址信息获取异常堆栈数据，获取侦测到的所述异常信号的信号类型；

将所述信号类型与所述异常堆栈数据组装后生成所述第一异常堆栈信息。

进一步地，所述异常堆栈信息获取程序被处理器执行时还实现如下操作：

读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息；

将所述第三异常堆栈信息与所述第二异常堆栈信息进行组装后上报至所述服务器，并在上报完成后退出所述项目。

需要说明的是，在本文中，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、物品或者装置不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、物品或者装置所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括该要素的过程、方法、物品或者装置中还存在另外的相同要素。

上述本发明实施例序号仅仅为了描述，不代表实施例的优劣。

通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到上述实施例方法可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件，但很多情况下前者是更佳的实施方式。基于这样的理解，本发明的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品存储在如上所述的一个存储介质(如 ROM/RAM、磁碟、光盘)中，包括若干指令用以使得一台终端设备(可以是手机，计算机，服务器，或者网络设备等)执行本发明各个实施例所述的方法。

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。

权 利 要 求 书

1、一种异常堆栈信息获取方法，其特征在于，所述方法包括：

在项目的运行过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取 native 层的第一异常堆栈信息；

读取侦测到所述异常信号时应用层记录的第二异常堆栈信息；

将所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目。

2、根据权利要求 1 所述的异常堆栈信息获取方法，其特征在于，所述在项目的运行过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取 native 层的第一异常堆栈信息的步骤包括：

在所述项目的运行过程中，当侦测到预设的异常信号时，确定运行所述项目的操作系统的版本信息；

调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息，并根据所述地址信息生成第一异常堆栈信息。

3、根据权利要求 2 所述的异常堆栈信息获取方法，其特征在于，所述调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息，并根据所述地址信息生成第一异常堆栈信息的步骤包括：

调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息；

根据所述地址信息获取异常堆栈数据，获取侦测到的所述异常信号的信号类型；

将所述信号类型与所述异常堆栈数据组装后生成所述第一异常堆栈信息。

4、根据权利要求 2 所述的异常堆栈信息获取方法，其特征在于，所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目的步骤包括：

读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息；

将所述第三异常堆栈信息与所述第二异常堆栈信息进行组装后上报至所述服务器，并在上报完成后退出所述项目。

5、根据权利要求 3 所述的异常堆栈信息获取方法，其特征在于，所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目的

步骤包括：

读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息；

将所述第三异常堆栈信息与所述第二异常堆栈信息进行组装后上报至所述服务器，并在上报完成后退出所述项目。

6、根据权利要求 3 所述的异常堆栈信息获取方法，其特征在于，所述读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息的步骤包括：

启动信息还原脚本，基于所述信息还原脚本，将所述第一异常堆栈信息中的函数名依次与所述符号表中的函数名进行一一匹配，获取到所述第一异常堆栈信息中的函数名对应的文件名和行号；

将获取的文件名和行号添加至原始的第一异常堆栈信息，生成所述服务器可读的第三异常堆栈信息。

7、根据权利要求 4 所述的异常堆栈信息获取方法，其特征在于，所述读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息的步骤包括：

启动信息还原脚本，基于所述信息还原脚本，将所述第一异常堆栈信息中的函数名依次与所述符号表中的函数名进行一一匹配，获取到所述第一异常堆栈信息中的函数名对应的文件名和行号；

将获取的文件名和行号添加至原始的第一异常堆栈信息，生成所述服务器可读的第三异常堆栈信息。

8、一种异常堆栈信息获取装置，其特征在于，所述装置包括：存储器、处理器及存储在所述存储器上并可在所述处理器上运行的异常堆栈信息获取程序，所述异常堆栈信息获取程序被所述处理器执行时实现如下步骤：

在项目的运行过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取 native 层的第一异常堆栈信息；

读取侦测到所述异常信号时应用层记录的第二异常堆栈信息；

将所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目。

9、根据权利要求 8 所述的异常堆栈信息获取装置，其特征在于，所述在项目的运行

过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取 native 层的第一异常堆栈信息的步骤包括：

在所述项目的运行过程中，当侦测到预设的异常信号时，确定运行所述项目的操作系统的版本信息；

调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息，并根据所述地址信息生成第一异常堆栈信息。

10、根据权利要求 8 所述的异常堆栈信息获取装置，其特征在于，所述调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息，并根据所述地址信息生成第一异常堆栈信息的步骤包括：

调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息；

根据所述地址信息获取异常堆栈数据，获取侦测到的所述异常信号的信号类型；

将所述信号类型与所述异常堆栈数据组装后生成所述第一异常堆栈信息。

11、根据权利要求 9 所述的异常堆栈信息获取装置，其特征在于，所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目的步骤包括：

读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息；

将所述第三异常堆栈信息与所述第二异常堆栈信息进行组装后上报至所述服务器，并在上报完成后退出所述项目。

12、根据权利要求 10 所述的异常堆栈信息获取装置，其特征在于，所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目的步骤包括：

读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息；

将所述第三异常堆栈信息与所述第二异常堆栈信息进行组装后上报至所述服务器，并在上报完成后退出所述项目。

13、根据权利要求 10 所述的异常堆栈信息获取装置，其特征在于，所述读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息的步骤包括：

启动信息还原脚本，基于所述信息还原脚本，将所述第一异常堆栈信息中的函数名依次与所述符号表中的函数名进行一一匹配，获取到所述第一异常堆栈信息中的函数名对应的文件名和行号；

将获取的文件名和行号添加至原始的第一异常堆栈信息，生成所述服务器可读的第三异常堆栈信息。

14、根据权利要求 11 所述的异常堆栈信息获取装置，其特征在于，所述读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息的步骤包括：

启动信息还原脚本，基于所述信息还原脚本，将所述第一异常堆栈信息中的函数名依次与所述符号表中的函数名进行一一匹配，获取到所述第一异常堆栈信息中的函数名对应的文件名和行号；

将获取的文件名和行号添加至原始的第一异常堆栈信息，生成所述服务器可读的第三异常堆栈信息。

15、一种计算机可读存储介质，其特征在于，所述计算机可读存储介质上存储有异常堆栈信息获取程序，所述异常堆栈信息获取程序被处理器执行时实现如下步骤：

在项目的运行过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取 native 层的第一异常堆栈信息；

读取侦测到所述异常信号时应用层记录的第二异常堆栈信息；

将所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目。

16、根据权利要求 15 所述的计算机可读存储介质，其特征在于，所述在项目的运行过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取 native 层的第一异常堆栈信息的步骤包括：

在所述项目的运行过程中，当侦测到预设的异常信号时，确定运行所述项目的操作系统的版本信息；

调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息，并根据所述地址信息生成第一异常堆栈信息。

17、根据权利要求 16 所述的计算机可读存储介质，其特征在于，所述调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息，并根据所述地址信息生成第一异常堆栈信息的步骤包括：

调用并执行与所述操作系统的版本信息对应的异常信号处理函数，获取发生异常的地址信息；

根据所述地址信息获取异常堆栈数据，获取侦测到的所述异常信号的信号类型；

将所述信号类型与所述异常堆栈数据组装后生成所述第一异常堆栈信息。

18、根据权利要求 16 所述的计算机可读存储介质，其特征在于，所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目的步骤包括：

读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息；

将所述第三异常堆栈信息与所述第二异常堆栈信息进行组装后上报至所述服务器，并在上报完成后退出所述项目。

19、根据权利要求 17 所述的计算机可读存储介质，其特征在于，所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目的步骤包括：

读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息；

将所述第三异常堆栈信息与所述第二异常堆栈信息进行组装后上报至所述服务器，并在上报完成后退出所述项目。

20、根据权利要求 17 所述的计算机可读存储介质，其特征在于，所述读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息的步骤包括：

启动信息还原脚本，基于所述信息还原脚本，将所述第一异常堆栈信息中的函数名依次与所述符号表中的函数名进行一一匹配，获取到所述第一异常堆栈信息中的函数名对应的文件名和行号；

将获取的文件名和行号添加至原始的第一异常堆栈信息，生成所述服务器可读的第三异常堆栈信息。

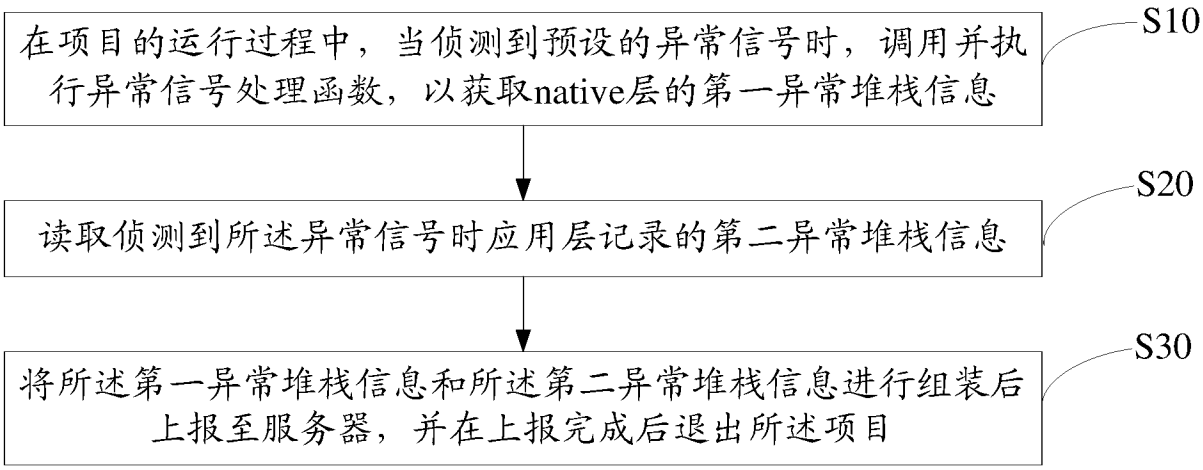
在项目的运行过程中，当侦测到预设的异常信号时，调用并执行异常信号处理函数，以获取native层的第一异常堆栈信息
S10
读取侦测到所述异常信号时应用层记录的第二异常堆栈信息
S20
将所述第一异常堆栈信息和所述第二异常堆栈信息进行组装后上报至服务器，并在上报完成后退出所述项目
S30

图 1

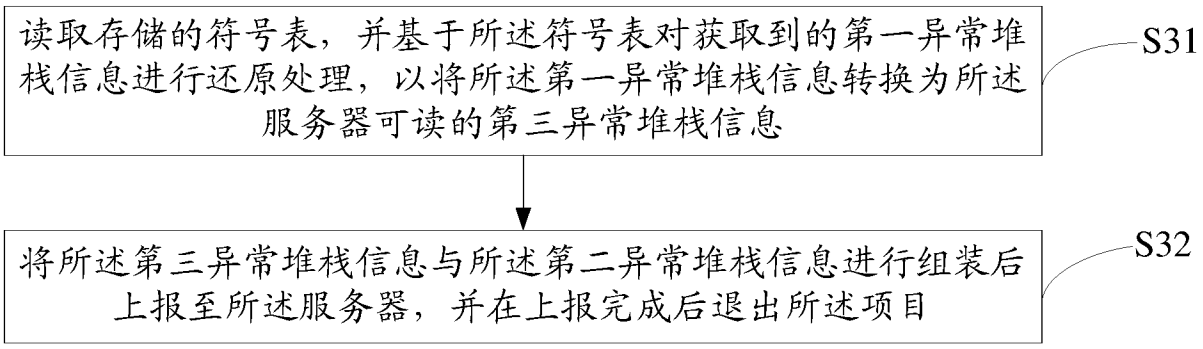
读取存储的符号表，并基于所述符号表对获取到的第一异常堆栈信息进行还原处理，以将所述第一异常堆栈信息转换为所述服务器可读的第三异常堆栈信息
S31
将所述第三异常堆栈信息与所述第二异常堆栈信息进行组装后上报至所述服务器，并在上报完成后退出所述项目
S32

图 2

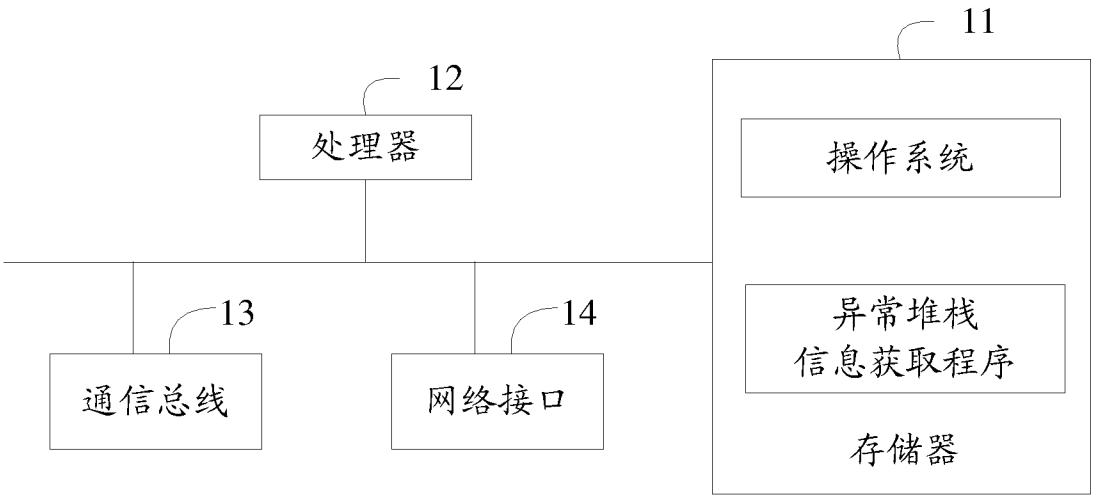
11
12
处理器
操作系统
13
14
异常堆栈
信息获取程序
通信总线
网络接口
存储器

图 3

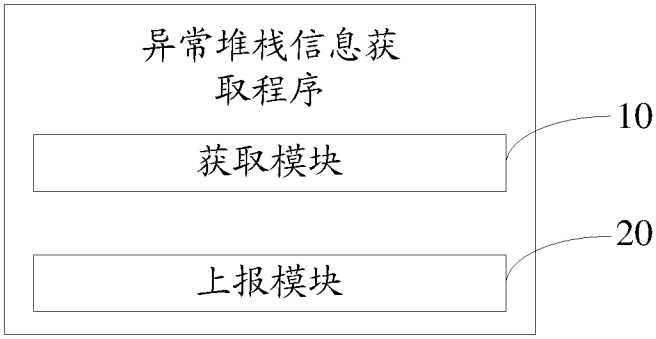
异常堆栈信息获
取程序
10
获取模块
20
上报模块

图 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2017/105018

A. CLASSIFICATION OF SUBJECT MATTER

G06F 11/07 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPI, EPODOC, CNKI, CNPAT, IEEE: 异常, 出错, 堆栈, 本地, 应用层, 服务器, 信号, 地址, native, bug, error, fault, app, application, stack, server, signal, address

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 104424090 A (TENCENT TECHNOLOGY (SHENZHEN) CO., LTD.), 18 March 2015 (18.03.2015), description, paragraphs [0041]-[0076]	1-20
A	US 2016299832 A1 (GEMALTO SA), 13 October 2016 (13.10.2016), entire document	1-20
A	CN 105446822 A (PHICOMM (SHANGHAI) CO., LTD.), 30 March 2016 (30.03.2016), entire document	1-20
A	WO 2009146612 A1 (HUAWEI TECHNOLOGIES CO., LTD.), 10 December 2009 (10.12.2009), entire document	1-20

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
07 February 2018	24 February 2018
Name and mailing address of the ISA State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No. (86-10) 62019451	Authorized officer YAO, Nan Telephone No. (86-10) 010-61648474

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2017/105018

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 104424090 A	18 March 2015	None	
US 2016299832 A1	13 October 2016	WO 2015078614 A1	04 June 2015
		EP 3074911 A1	05 October 2016
		EP 2879074 A1	03 June 2015
CN 105446822 A	30 March 2016	None	
WO 2009146612 A1	10 December 2009	CN 101599039 A	09 December 2009
		CN 101599039 B	02 November 2011

国际检索报告

国际申请号
PCT/CN2017/105018

A. 主题的分类

G06F 11/07(2006.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用））

WPI,EPODOC,CNKI,CNPAT,IEEE:异常,出错,堆栈,本地,应用层,服务器,信号,地址,native,bug,error,fault,app,application,stack,server,signal,address

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
A	CN 104424090 A (腾讯科技深圳有限公司) 2015年 3月 18日 (2015-03-18) 说明书第[0041]-[0076]段	1-20
A	US 2016299832 A1 (GEMALTO SA) 2016年 10月 13日 (2016-10-13) 全文	1-20
A	CN 105446822 A (上海斐讯数据通信技术有限公司) 2016年 3月 30日 (2016-03-30) 全文	1-20
A	WO 2009146612 A1 (华为技术有限公司) 2009年 12月 10日 (2009-12-10) 全文	1-20

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2018年 2月 7日	2018年 2月 24日
ISA/CN的名称和邮寄地址 中华人民共和国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10)62019451	受权官员 姚楠 电话号码 (86-10)010-61648474

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2017/105018

检索报告引用的专利文件			公布日（年/月/日）	同族专利			公布日（年/月/日）
CN	104424090	A	2015年 3月 18日	无			
US	2016299832	A1	2016年 10月 13日	WO	2015078614	A1	2015年 6月 4日
				EP	3074911	A1	2016年 10月 5日
				EP	2879074	A1	2015年 6月 3日
CN	105446822	A	2016年 3月 30日	无			
WO	2009146612	A1	2009年 12月 10日	CN	101599039	A	2009年 12月 9日
				CN	101599039	B	2011年 11月 2日