

(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
2020 年 10 月 29 日 (29.10.2020)



(10) 国际公布号
WO 2020/215567 A1

(51) 国际专利分类号:
G06F 11/36 (2006.01)

(21) 国际申请号: PCT/CN2019/103162

(22) 国际申请日: 2019 年 8 月 29 日 (29.08.2019)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201910341769.2 2019 年 4 月 26 日 (26.04.2019) 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福田街道福安社区益田路 5033 号平安金融中心 23 楼, Guangdong 518033 (CN)。

(72) 发明人: 何兵(HE, Bing); 中国广东省深圳市福田区福田街道福安社区益田路 5033 号平安金融中心 23 楼, Guangdong 518033 (CN)。

(74) 代理人: 北京市京大律师事务所(BEIJING JINGDA LAW FIRM); 中国北京市海淀区海淀中街 16 号 7 层 2 单元 703, Beijing 100080 (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: GLOBAL HOOK AUTOMATIC REPAIR METHOD, APPARATUS, DEVICE, AND STORAGE MEDIUM

(54) 发明名称: 全局hook自动修复方法、装置、设备及存储介质

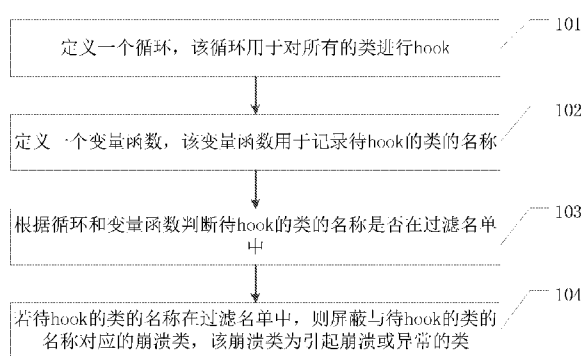


图 1

101 Define loop, loop being used for hooking all classes
102 Define variable function, variable function being used for recording name of class to be hooked
103 According to loop and variable function, determine whether name of class to be hooked is in filter list
104 If name of class to be hooked is in filter list, then block crash class corresponding to name of class to be hooked, crash class being category which causes crash or exception

(57) Abstract: Disclosed by the present application are a global hook automatic repair method, apparatus, device, and storage medium based on the field of security monitoring, used for automatically blocking classes which cause global crashes, avoiding modification of the code and improving repair efficiency. The method of the present application comprises: defining a loop, said loop being used for hooking all classes; defining a variable function, said variable function being used for recording the name of a class to be hooked; according to the loop and the variable function, determining whether the name of the class to be hooked is in a filter list; if the name of the class to be hooked is in said filter list, then blocking the crash class corresponding to the name of the class to be hooked, said crash class being a category which causes a crash or an exception.

(57) 摘要: 本申请公开了基于安全监控领域的全局hook自动修复方法、装置、设备及存储介质, 用于自动屏蔽引起全局崩溃的类, 避免了对代码的修改, 提高了修复效率。本申请方法包括: 定义一个循环, 所述循环用于对所有的类进行hook; 定义一个变量函数, 所述变量函数用于记录待hook的类的名称; 根据所述循环和所述变量函数判断所述待hook的类的名称是否在过滤名单中; 若所述待hook的类的名称在所述过滤名单中, 则屏蔽与所述待hook的类的名称对应的崩溃类, 所述崩溃类为引起崩溃或异常的类。

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))。

— 1 —

全局hook自动修复方法、装置、设备及存储介质

本申请要求于2019年4月26日提交中国专利局、申请号为201910341769.2、发明名称为“全局hook自动修复方法、装置、设备及存储介质”的中国专利申请的优先权，其全部内容通过引用结合在申请中。

5

技术领域

本发明涉及操作系统安全监控领域，尤其涉及全局hook自动修复方法、装置、设备及存储介质。

背景技术

10 随着手机行业的快速发展，手机系统也相应的发展，目前主流的系统有安卓系统和 IOS 系统。对于 IOS 系统而言，IOS 项目如果需要测试代码覆盖率，一个比较好的办法就是 hook 所有的类，然后在所有类的所有方法中添加统计代码，但是因为各个项目的类的种类复杂和数量庞大，有些类的写法会导致一 hook 就引起崩溃。

15 目前比较常规的做法是通过人工进行手动排除，发明人意识到需要人工修改代码进行屏蔽引起崩溃的相关的类，修复效率低。

发明内容

本申请实施例提供了一种全局 hook 自动修复方法、装置、设备及存储介质，用于自动屏蔽引起全局崩溃的类，避免了对代码的修改，提高了修复效率。

20 本申请实施例的第一方面提供一种基于数据分析的全局 hook 自动修复方法，包括：定义一个循环，所述循环用于对所有的类进行 hook；定义一个变量函数，所述变量函数用于记录待 hook 的类的名称；根据所述循环和所述变量函数判断所述待 hook 的类的名称是否在过滤名单中；若所述待 hook 的类的名称在所述过滤名单中，则屏蔽与所述待 hook 的类的名称对应的崩溃类，所述崩溃类为引起崩溃或异常的类。

25 可选的，在本申请实施例第一方面的第一种实现方式中，所述根据所述循环和所述变量函数判断所述待 hook 的类的名称是否在过滤名单中之后，所述方法还包括：若所述待 hook 的类的名称不在所述过滤名单中，则调用所述循环对所述待 hook 的类进行 hook。

30 可选的，在本申请实施例第一方面的第二种实现方式中，所述屏蔽与所述待 hook 的类的名称对应的崩溃类，所述崩溃类为引起崩溃或异常的类之后，所述方法还包括：监听异常函数，所述异常函数用于监控并记录所述崩溃类对应的数组；根据所述崩溃类对应的数组确定所述异常函数中的崩溃类的名称；将所述崩溃类的名称导入到所述过滤名单。

—2—

可选的，在本申请实施例第一方面的第三种实现方式中，所述定义一个循环，所述循环用于对所有的类进行 hook 包括：获取当前运行的目标应用程序的路径；获取所述目标应用程序的各个类的名称以及数量；根据所述各个类的名称以及数量定义一个 for 循环，所述 for 循环用于按照所述目标应用程序的路径对各个类进行全局 hook。

5 可选的，在本申请实施例第一方面的第四种实现方式中，所述定义一个变量函数，所述变量函数用于记录待 hook 的类的名称包括：获取预置的 key 值；根据所述预置的 key 值调用存储于系统存储器或者预置数据库中的崩溃类的数组；根据所述崩溃类的数组定义一个变量函数，所述变量函数用于记录待 hook 的类的名称。

10 可选的，在本申请实施例第一方面的第五种实现方式中，所述根据所述循环和所述变量函数判断所述待 hook 的类的名称是否在过滤名单中包括：确定所述循环中调用的数组；在所述调用的数组中确定所述变量函数；确定所述变量函数中待 hook 的类的名称，并判断所述待 hook 的类的名称是否在过滤名单中。

15 可选的，在本申请实施例第一方面的第六种实现方式中，所述方法还包括：若所述过滤名单中崩溃类的数量大于阈值，则对引起崩溃的类进行整合；确定所述崩溃类中相同类型的崩溃类的前缀；将具有所述前缀的崩溃类进行批量屏蔽。

20 本申请实施例的第二方面提供了一种全局 hook 自动修复装置，包括：第一定义单元，用于定义一个循环，所述循环用于对所有的类进行 hook；第二定义单元，用于定义一个变量函数，所述变量函数用于记录待 hook 的类的名称；判断单元，用于根据所述循环和所述变量函数判断所述待 hook 的类的名称是否在过滤名单中；第一屏蔽单元，若所述待 hook 的类的名称在所述过滤名单中，则用于屏蔽与所述待 hook 的类的名称对应的崩溃类，所述崩溃类为引起崩溃或异常的类。

可选的，在本申请实施例第二方面的第一种实现方式中，全局 hook 自动修复装置还包括：调用单元，若所述待 hook 的类的名称不在所述过滤名单中，则用于调用所述循环对所述待 hook 的类进行 hook。

25 可选的，在本申请实施例第二方面的第二种实现方式中，全局 hook 自动修复装置还包括：监听单元，用于监听异常函数，所述异常函数用于监控并记录所述崩溃类对应的数组；第一确定单元，用于根据所述崩溃类对应的数组确定所述异常函数中的崩溃类的名称；导入单元，用于将所述崩溃类的名称导入到所述过滤名单。

可选的，在本申请实施例第二方面的第三种实现方式中，第一定义单元具体用于：获

—3—

取当前运行的目标应用程序的路径；获取所述目标应用程序的各个类的名称以及数量；根据所述各个类的名称以及数量定义一个 for 循环，所述 for 循环用于按照所述目标应用程序的路径对各个类进行全局 hook。

可选的，在本申请实施例第二方面的第四种实现方式中，第二定义单元具体用于：获取预置的 key 值；根据所述预置的 key 值调用存储于系统存储器或者预置数据库中的崩溃类的数组；根据所述崩溃类的数组定义一个变量函数，所述变量函数用于记录待 hook 的类的名称。

可选的，在本申请实施例第二方面的第五种实现方式中，判断单元具体用于：确定所述循环中调用的数组；在所述调用的数组中确定所述变量函数；确定所述变量函数中待 hook 的类的名称，并判断所述待 hook 的类的名称是否在过滤名单中。

可选的，在本申请实施例第二方面的第六种实现方式中，全局 hook 自动修复装置还包括：整合单元，若所述过滤名单中崩溃类的数量大于阈值，则用于对引起崩溃的类进行整合；第二确定单元，用于确定所述崩溃类中相同类型的崩溃类的前缀；第二屏蔽单元，用于将具有所述前缀的崩溃类进行批量屏蔽。

本申请实施例的第三方面提供了一种全局 hook 自动修复设备，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计算机程序，所述处理器执行所述计算机程序时实现上述任一实施方式所述的全局 hook 自动修复方法。

本申请实施例的第四方面提供了一种计算机可读存储介质，所述计算机可读存储介质中存储有指令，当所述指令在计算机上运行时，使得计算机执行上述任一实施方式所述的全局 hook 自动修复方法的步骤。

本申请实施例提供的技术方案中，定义一个循环，该循环用于对所有的类进行 hook；定义一个变量函数，该变量函数用于记录待 hook 的类的名称；根据循环和变量函数判断待 hook 的类的名称是否在过滤名单中；若待 hook 的类的名称在过滤名单中，则屏蔽与待 hook 的类的名称对应的崩溃类，该崩溃类为引起崩溃或异常的类。本申请实施例，自动屏蔽引起全局崩溃的类，避免了对代码的修改，提高了修复效率。

附图说明

图 1 为本申请实施例中全局 hook 自动修复方法的一个实施例示意图；

图 2 为本申请实施例中全局 hook 自动修复方法的另一个实施例示意图；

图 3 为本申请实施例中全局 hook 自动修复装置的一个实施例示意图；

图 4 为本申请实施例中全局 hook 自动修复装置的另一个实施例示意图；

图 5 为本申请实施例中全局 hook 自动修复设备的一个实施例示意图。

具体实施方式

本申请实施例提供了一种全局 hook 自动修复方法、装置、设备及存储介质，用于自动
5 屏蔽引起全局崩溃的类，避免了对代码的修改，提高了修复效率。

为了使本技术领域的人员更好地理解本申请方案，下面将结合本申请实施例中的附图，
对本申请实施例进行描述。

本申请的说明书和权利要求书及上述附图中的术语“第一”、“第二”、“第三”、“第四”
等（如果存在）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
10 解这样使用的数据在适当情况下可以互换，以便这里描述的实施例能够以除了在这里图示
或描述的内容以外的顺序实施。此外，术语“包括”或“具有”及其任何变形，意图在于
覆盖不排除他的包含，例如，包含了一系列步骤或单元的过程、方法、系统、产品或设备不
必限于清楚地列出的那些步骤或单元，而是可包括没有清楚地列出的或对于这些过程、方
法、产品或设备固有的其它步骤或单元。

15 请参阅图 1，本申请实施例提供一种全局 hook 自动修复方法的流程图，具体包括：

101、定义一个循环，该循环用于对所有的类进行 hook。

服务器定义一个循环，该循环用于对所有的类进行 hook。具体的，服务器定义一个 for
循环，通过该 for 循环对系统或应用程序的所有类进行 hook。例如，服务器获取当前运行
的目标应用程序的路径；服务器获取目标应用程序的各个类的名称以及数量；服务器根据
20 各个类的名称以及数量定义一个 for 循环，for 循环用于按照目标应用程序的路径对各个
类进行全局 hook。

例如，当需要对一个应用程序（application，APP）进行 hook 时，首先获取当前运行
的 APP 的路径，如，`dladdr(&_mh_execute_header, &info)`；其次获取当前运行的 APP 的
各个类的名称，及各个类的数量。例如，`classes =`
25 `objc_copyClassNamesForImage(info.dli_fname, &count)`；其中，二维数组 `classes` 存放
有所有类的名称，`count` 存储有所有的类的数量；最后根据各个类的名称以及数量定义一
个 for 循环。例如，将数组 `classes` 数组中的类一个一个取出来，for 循环开始。

102、定义一个变量函数，该变量函数用于记录待 hook 的类的名称。

服务器定义一个变量函数，该变量函数用于记录待 hook 的类的名称。

—5—

具体的，服务器先获取预置的 key 值；然后根据预置的 key 值调用崩溃类的数组，崩溃类的数组存储于系统存储器或者预置数据库中；再根据崩溃类的数组定义一个变量函数，其中，该变量函数用于记录待 hook 的类的名称。

例如，服务器定义的一个字符串，比如就叫 PA_HookCrashListKey，然后使用系统自带的存储器（或者预先写好的数据库代码），进行存储和调用。崩溃的类数组就是，通过 key 从数据库中获取的对象。

103、根据循环和变量函数判断待 hook 的类的名称是否在过滤名单中。

服务器根据循环和变量函数判断待 hook 的类的名称是否在过滤名单中。具体的，服务器确定循环中调用的数组；服务器在调用的数组中确定变量函数；服务器确定变量函数中待 hook 的类的名称，并判断待 hook 的类的名称是否在过滤名单中。若待 hook 的类的名称在过滤名单中，则服务器屏蔽与待 hook 的类的名称对应的崩溃类。若待 hook 的类的名称不在过滤名单中，则服务器调用循环对待 hook 的类进行 hook。

具体的，服务器通过 for 循环对待 hook 的类进行 hook，其中，钩子(hook)是 Windows 消息处理机制的一个平台, 应用程序 (APP) 可以在上面设置子程以监视指定窗口的某种消息，而且所监视的窗口可以是其他进程所创建的。当消息到达后，在目标窗口处理函数之前处理它。钩子机制允许应用程序截获处理 window 消息或特定事件。钩子实际上是一个处理消息的程序段，通过系统调用，把它挂入系统。每当特定的消息发出后，且在没有到达目的窗口前，钩子程序就先捕获该消息，即钩子函数先得到控制权。这时钩子函数即可以加工处理（改变）该消息，也可以不作处理而继续传递该消息，还可以强制结束消息的传递。

需要说明的是，hook 的顺序是随机的，取决于系统方法返回给我的所有类的数组顺序，我再从数组中一个一个拿出来进行 hook。而 hook 标准为：获取类的所有方法，过滤一些系统自带的方法和需要过滤的方法，需要过滤的方法取决于项目和业务，是工作人员预先配置的，而对剩下的方法，将方法地址更改，强行触发面向对象的 C (objective-C, OC) 语言的消息转发机制，然后再重写消息转发时调用的 forwardInvocation 方法，在该方法里面将地址换回去，但是在换回去之前，还可以写一些 hook 代码，从而达到 hook 的目的。

104、若待 hook 的类的名称在过滤名单中，则屏蔽与待 hook 的类的名称对应的崩溃类，该崩溃类为引起崩溃或异常的类。

若待 hook 的类的名称在过滤名单中，则服务器屏蔽与待 hook 的类的名称对应的崩溃

类，该崩溃类为引起崩溃或异常的类。

可以理解的是，系统通过自带的异常函数进行监控，在每次崩溃之前，将数组保存到过滤名单范围。在下一次启动时，过滤名单会增加一个崩溃类的名称，并用来作为判断标准。

5 本申请实施例，定义一个循环，该循环用于对所有的类进行 hook；定义一个变量函数，该变量函数用于记录待 hook 的类的名称；根据循环和变量函数判断待 hook 的类的名称是否在过滤名单中；若待 hook 的类的名称在过滤名单中，则屏蔽与待 hook 的类的名称对应的崩溃类，该崩溃类为引起崩溃或异常的类。可以自动屏蔽引起全局崩溃的类，避免了对系统代码的修改，提高了修复效率。

10 请参阅图 2，本申请实施例中全局 hook 自动修复方法的另一个实施例包括：

201、定义一个循环，该循环用于对所有的类进行 hook。

服务器定义一个循环，该循环用于对所有的类进行 hook。具体的，服务器定义一个 for 循环，通过该 for 循环对系统或应用程序的所有类进行 hook。例如，服务器获取当前运行的目标应用程序的路径；服务器获取目标应用程序的各个类的名称以及数量；服务器根据
15 各个类的名称以及数量定义一个 for 循环，for 循环用于按照目标应用程序的路径对各个类进行全局 hook。

例如，当需要对一个应用程序（application，APP）进行 hook 时，首先获取当前运行的 APP 的路径，如，`dladdr(&_mh_execute_header, &info)`；其次获取当前运行的 APP 的各个类的名称，及各个类的数量。例如，`classes =
20 objc_copyClassNamesForImage(info.dli_fname, &count)`；其中，二维数组 `classes` 存放有所有类的名称，`count` 存储有所有的类的数量；最后根据各个类的名称以及数量定义一个 for 循环。例如，将数组将 `classes` 数组中的类一个一个取出来，for 循环开始。

202、定义一个变量函数，该变量函数用于记录待 hook 的类的名称。

服务器定义一个变量函数，该变量函数用于记录待 hook 的类的名称。

25 具体的，服务器先获取预置的 key 值；然后根据预置的 key 值调用崩溃类的数组，崩溃类的数组存储于系统存储器或者预置数据库中；再根据崩溃类的数组定义一个变量函数，其中，该变量函数用于记录待 hook 的类的名称。

例如，服务器定义的一个字符串，比如就叫 `PA_HookCrashListKey`，然后使用系统自带的存储器（或者预先写好的数据库代码），进行存储和调用。崩溃的类数组就是，通过

key 从数据库中获取的对象。

203、根据循环和变量函数判断待 hook 的类的名称是否在过滤名单中。

服务器根据循环和变量函数判断待 hook 的类的名称是否在过滤名单中。具体的，服务器确定循环中调用的数组；服务器在调用的数组中确定变量函数；服务器确定变量函数中待 hook 的类的名称，并判断待 hook 的类的名称是否在过滤名单中。若待 hook 的类的名称在过滤名单中，则服务器屏蔽与待 hook 的类的名称对应的崩溃类。若待 hook 的类的名称不在过滤名单中，则服务器调用循环对待 hook 的类进行 hook。

具体的，服务器通过 for 循环对待 hook 的类进行 hook，其中，钩子(hook)是 Windows 消息处理机制的一个平台, 应用程序 (APP) 可以在上面设置子程以监视指定窗口的某种消息，而且所监视的窗口可以是其他进程所创建的。当消息到达后，在目标窗口处理函数之前处理它。钩子机制允许应用程序截获处理 window 消息或特定事件。钩子实际上是一个处理消息的程序段，通过系统调用，把它挂入系统。每当特定的消息发出后，且在没有到达目的窗口前，钩子程序就先捕获该消息，亦即钩子函数先得到控制权。这时钩子函数即可以加工处理（改变）该消息，也可以不作处理而继续传递该消息，还可以强制结束消息的传递。

需要说明的是，hook 的顺序是随机的，取决于系统方法返回给我的所有类的数组顺序，我再从数组中一个一个拿出来进行 hook。而 hook 标准为：获取类的所有方法，过滤一些系统自带的方法和需要过滤的方法，需要过滤的方法取决于项目和业务，是工作人员预先配置的，而对剩下的方法，将方法地址更改，强行触发面向 OC 语言的消息转发机制，然后再重写消息转发时调用的 forwardInvocation 方法，在该方法里面将地址换回去，但是在换回去之前，还可以写一些 hook 代码，从而达到 hook 的目的。

204、若待 hook 的类的名称在过滤名单中，则屏蔽与待 hook 的类的名称对应的崩溃类，该崩溃类为引起崩溃或异常的类。

若待 hook 的类的名称在过滤名单中，则服务器屏蔽与待 hook 的类的名称对应的崩溃类，该崩溃类为引起崩溃或异常的类。

可以理解的是，系统通过自带的异常函数进行监控，在每次崩溃之前，将数组保存到过滤名单范围。在下一次启动时，过滤名单会增加一个崩溃类的名称，并将增加的崩溃类的名称作为判断标准。

205、监听异常函数，该异常函数用于监控并记录崩溃类对应的数组。

服务器对系统中的异常函数进行监听，其中，该异常函数用于监控并记录崩溃类对应的数组。数组存储有与各个类对应的具体数据。例如，若变量函数为 a，因为数组中的 a，是在开始 hook 之前存进去的，即在数据中做一个标记，然后在 hook 代码的方法调用之后，把 a 删掉。反过来，如果 hook 代码崩溃了，后面的代码没机会运行，这个数组中的 a 就不会被删掉，下次 hook 的时候，发现这个 a 在崩溃数组中，我们就不再 hook 了，避免陷入循环崩溃。

206、根据崩溃类对应的数组确定异常函数中的崩溃类的名称。

服务器根据崩溃类对应的数组确定异常函数中的崩溃类的名称。例如，服务器根据崩溃类对应的数组在异常函数 `NSSetUncaughtExceptionHandler` 中确定发生崩溃的崩溃类的名称。

207、将崩溃类的名称导入到过滤名单。

服务器将崩溃类的名称导入到过滤名单，该过滤名单中包括会引起系统崩溃或异常的类的名称。

可选的，若过滤名单中崩溃类的数量大于阈值，则服务器对引起崩溃的类进行整合；服务器确定崩溃类中相同类型的崩溃类的前缀；服务器将具有前缀的崩溃类进行批量屏蔽。

本申请实施例，定义一个循环，该循环用于对所有的类进行 hook；定义一个变量函数，该变量函数用于记录待 hook 的类的名称；根据循环和变量函数判断待 hook 的类的名称是否在过滤名单中；若待 hook 的类的名称在过滤名单中，则屏蔽与待 hook 的类的名称对应的崩溃类，该崩溃类为引起崩溃或异常的类；监听异常函数，该异常函数用于监控并记录崩溃类对应的数组。本申请实施例中，系统每崩溃一次，崩溃数组中就会多一个需要屏蔽的类，下次启动 hook 代码的时候，按照前面的策略，对这个类不执行 hook，这样崩溃过的类就不会 hook 了，也就不会导致系统崩溃了，可以自动屏蔽引起全局崩溃的类，避免了对系统代码的修改，提高了修复效率。

可选的，在上述图 2 对应的实施例的基础上，本申请实施例提供的全局 hook 自动修复方法的可选实施例中，还可以包括：

判断循环中变量函数是否为空；若变量函数不为空，则将崩溃类的数组中与变量函数对应的类删除。

具体的，服务器判断循环中变量函数是否为空。若变量函数不为空，则服务器将崩溃类的数组中与变量函数对应的类删除。若变量函数为空，则服务器忽略该变量函数。

可选的，在上述图 2 对应的实施例的基础上，本申请实施例提供的全局 hook 自动修复方法的可选实施例中，方法还包括：

若所述待 hook 的类的名称不在所述过滤名单中，则通过循环对待 hook 的类进行 hook。

本申请实施例，定义一个循环，该循环用于对所有的类进行 hook；定义一个变量函数，该变量函数用于记录待 hook 的类的名称；根据循环和变量函数判断待 hook 的类的名称是否在过滤名单中；若待 hook 的类的名称在过滤名单中，则屏蔽与待 hook 的类的名称对应的崩溃类，该崩溃类为引起崩溃或异常的类；监听异常函数，该异常函数用于监控并记录崩溃类对应的数组。可以自动屏蔽引起全局崩溃的类，避免了对系统代码的修改，提高了修复效率。

上面对本申请实施例中全局 hook 自动修复方法进行了描述，下面对本申请实施例中全局 hook 自动修复装置进行描述，请参阅图 3，本申请实施例中全局 hook 自动修复装置的一个实施例包括：

第一定义单元 301，用于定义一个循环，所述循环用于对所有的类进行 hook；

第二定义单元 302，用于定义一个变量函数，所述变量函数用于记录待 hook 的类的名称；

判断单元 303，用于根据所述循环和所述变量函数判断所述待 hook 的类的名称是否在过滤名单中；

第一屏蔽单元 304，若所述待 hook 的类的名称在所述过滤名单中，则用于屏蔽与所述待 hook 的类的名称对应的崩溃类，所述崩溃类为引起崩溃或异常的类。

本申请实施例，定义一个循环，该循环用于对所有的类进行 hook；定义一个变量函数，该变量函数用于记录待 hook 的类的名称；根据循环和变量函数判断待 hook 的类的名称是否在过滤名单中；若待 hook 的类的名称在过滤名单中，则屏蔽与待 hook 的类的名称对应的崩溃类，该崩溃类为引起崩溃或异常的类。可以自动屏蔽引起全局崩溃的类，避免了对系统代码的修改，提高了修复效率。

请参阅图 4，本申请实施例中全局 hook 自动修复装置的一个实施例包括：

第一定义单元 301，用于定义一个循环，所述循环用于对所有的类进行 hook；

第二定义单元 302，用于定义一个变量函数，所述变量函数用于记录待 hook 的类的名称；

判断单元 303，用于根据所述循环和所述变量函数判断所述待 hook 的类的名称是否在

过滤名单中；

第一屏蔽单元 304，若所述待 hook 的类的名称在所述过滤名单中，则用于屏蔽与所述待 hook 的类的名称对应的崩溃类，所述崩溃类为引起崩溃或异常的类。

可选的，全局 hook 自动修复装置还包括：

- 5 调用单元 305，若所述待 hook 的类的名称不在所述过滤名单中，则用于通过所述循环对所述待 hook 的类进行 hook。

可选的，全局 hook 自动修复装置还包括：

监听单元 306，用于监听异常函数，所述异常函数用于监控并记录所述崩溃类对应的数组；

- 10 第一确定单元 307，用于根据所述崩溃类对应的数组确定所述异常函数中的崩溃类的名称；

导入单元 308，用于将所述崩溃类的名称导入到所述过滤名单。

可选的，第一定义单元 301 具体用于：

- 15 获取当前运行的目标应用程序的路径；获取所述目标应用程序的各个类的名称以及数量；根据所述各个类的名称以及数量定义一个 for 循环，所述 for 循环用于按照所述目标应用程序的路径对各个类进行全局 hook。

可选的，第二定义单元 302 具体用于：

- 20 获取预置的 key 值；根据所述预置的 key 值调用存储于系统存储器或者预置数据库中的崩溃类的数组；根据所述崩溃类的数组定义一个变量函数，所述变量函数用于记录待 hook 的类的名称。

可选的，判断单元 303 具体用于：

确定所述循环中调用的数组；在所述调用的数组中确定所述变量函数；确定所述变量函数中待 hook 的类的名称，并判断所述待 hook 的类的名称是否在过滤名单中。

可选的，全局 hook 自动修复装置还包括：

- 25 整合单元 309，若所述过滤名单中崩溃类的数量大于阈值，则用于对引起崩溃的类进行整合；

第二确定单元 310，用于确定所述崩溃类中相同类型的崩溃类的前缀；

第二屏蔽单元 311，用于将具有所述前缀的崩溃类进行批量屏蔽。

本申请实施例，定义一个循环，该循环用于对所有的类进行 hook；定义一个变量函数，

该变量函数用于记录待 hook 的类的名称；根据循环和变量函数判断待 hook 的类的名称是否在过滤名单中；若待 hook 的类的名称在过滤名单中，则屏蔽与待 hook 的类的名称对应的崩溃类，该崩溃类为引起崩溃或异常的类；监听异常函数，该异常函数用于监控并记录崩溃类对应的数组。可以自动屏蔽引起全局崩溃的类，避免了对系统代码的修改，提高了修复效率。

上面图 3 至图 4 从模块化功能实体的角度对本申请实施例中的全局 hook 自动修复装置进行详细描述，下面从硬件处理的角度对本申请实施例中全局 hook 自动修复设备进行详细描述。

图 5 是本申请实施例提供的一种全局 hook 自动修复设备的结构示意图，该全局 hook 自动修复设备 500 可因配置或性能不同而产生比较大的差异，可以包括一个或一个以上处理器（central processing units, CPU）501（例如，一个或一个以上处理器）和存储器 509，一个或一个以上存储应用程序 507 或数据 506 的存储介质 508（例如一个或一个以上海量存储设备）。其中，存储器 509 和存储介质 508 可以是短暂存储或持久存储。存储在存储介质 508 的程序可以包括一个或一个以上模块（图示没标出），每个模块可以包括对全局 hook 自动修复设备中的一系列指令操作。更进一步地，处理器 501 可以设置为与存储介质 508 通信，在全局 hook 自动修复设备 500 上执行存储介质 508 中的一系列指令操作。

全局 hook 自动修复设备 500 还可以包括一个或一个以上电源 502，一个或一个以上有线或无线网络接口 503，一个或一个以上输入输出接口 504，和/或，一个或一个以上操作系统 505，例如 Windows Serve, Mac OS X, Unix, Linux, FreeBSD 等等。本领域技术人员可以理解，图 5 中示出的全局 hook 自动修复设备结构并不构成对全局 hook 自动修复设备的限定，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。处理器 501 可以执行上述实施例中第一定义单元 301、第二定义单元 302、判断单元 303、第一屏蔽单元 304、调用单元 305、监听单元 306、第一确定单元 307 和导入单元 308 中任一单元的功能。处理器 501 还可以执行上述实施例中整合单元 309、第二确定单元 310 和第二屏蔽单元 311 中任一单元的功能。

下面结合图 5 对全局 hook 自动修复设备的各个构成部件进行具体的介绍：

处理器 501 是全局 hook 自动修复设备的控制中心，可以按照设置的全局 hook 自动修复方法进行处理。处理器 501 利用各种接口和线路连接整个全局 hook 自动修复设备的各个部分，通过运行或执行存储在存储器 509 内的软件程序和/或模块，以及调用存储在存储器

509 内的数据，执行全局 hook 自动修复设备的各种功能和处理数据，从而实现崩溃类的屏蔽。存储介质 508 和存储器 509 都是存储数据的载体，本申请实施例中，存储介质 508 可以是指储存容量较小，但速度快的内存储器，而存储器 509 可以是储存容量大，但储存速度慢的外存储器。

5 存储器 509 可用于存储软件程序以及模块，处理器 501 通过运行存储在存储器 509 的软件程序以及模块，从而执行全局 hook 自动修复设备 500 的各种功能应用以及数据处理。存储器 509 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（比如监听异常函数等）等；存储数据区可存储根据全局 hook 自动修复设备的使用所创建的数据（比如定义一个变量函数等）等。此外，存储器 509 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、或其他非易失性固态存储器件。在本申请实施例中提供的全局 hook 自动修复方法程序和接收到的数据流存储在存储器中，当需要使用时，处理器 501 从存储器 509 中调用。

15 本申请还提供一种计算机可读存储介质，该计算机可读存储介质可以为非易失性计算机可读存储介质，所述计算机可读存储介质中存储有指令，当所述指令在计算机上运行时，使得计算机执行如下全局 hook 自动修复方法的步骤：

定义一个循环，所述循环用于对所有的类进行 hook；

定义一个变量函数，所述变量函数用于记录待 hook 的类的名称；

根据所述循环和所述变量函数判断所述待 hook 的类的名称是否在过滤名单中；

20 若所述待 hook 的类的名称在所述过滤名单中，则屏蔽与所述待 hook 的类的名称对应的崩溃类，所述崩溃类为引起崩溃或异常的类。

在计算机上加载和执行所述计算机程序指令时，全部或部分地产生按照本申请实施例所述的流程或功能。所述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。所述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计算机可读存储介质传输，例如，所述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、双绞线）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。所述计算机可读存储介质可以是计算机能够存储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。所述可用介质可以是磁性介质，（例如，软盘、硬

盘、磁带)、光介质(例如,光盘)、或者半导体介质(例如固态硬盘(Solid State Disk, SSD))等。

所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统,装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

5 在本申请所提供的几个实施例中,应该理解到,所揭露的系统,装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

10 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

15 另外,在本申请实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

20 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。基于这样的理解,本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或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一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(read-only memory, ROM)、随机存取存储器(random access memory, RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

25 以上所述,以上实施例仅用以说明本申请的技术方案,而非对其限制;尽管参照前述实施例对本申请进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围。

- 14 -

权 利 要 求

1、一种全局 hook 自动修复方法，包括：

定义一个循环，所述循环用于对所有的类进行 hook；

定义一个变量函数，所述变量函数用于记录待 hook 的类的名称；

5 根据所述循环和所述变量函数判断所述待 hook 的类的名称是否在过滤名单中；

若所述待 hook 的类的名称在所述过滤名单中，则屏蔽与所述待 hook 的类的名称对应的崩溃类，所述崩溃类为引起崩溃或异常的类。

2、根据权利要求 1 所述的全局 hook 自动修复方法，所述根据所述循环和所述变量函数判断所述待 hook 的类的名称是否在过滤名单中之后，所述方法还包括：

10 若所述待 hook 的类的名称不在所述过滤名单中，则调用所述循环对所述待 hook 的类进行 hook。

3、根据权利要求 2 所述的全局 hook 自动修复方法，所述屏蔽与所述待 hook 的类的名称对应的崩溃类，所述崩溃类为引起崩溃或异常的类之后，所述方法还包括：

15 监听异常函数，所述异常函数用于监控并记录所述崩溃类对应的数组；

根据所述崩溃类对应的数组确定所述异常函数中的崩溃类的名称；

将所述崩溃类的名称导入到所述过滤名单。

20 4、根据权利要求 1 所述的全局 hook 自动修复方法，所述定义一个循环，所述循环用于对所有的类进行 hook 包括：

获取当前运行的目标应用程序的路径；

获取所述目标应用程序的各个类的名称以及数量；

25 根据所述各个类的名称以及数量定义一个 for 循环，所述 for 循环用于按照所述目标应用程序的路径对各个类进行全局 hook。

5、根据权利要求 1 所述的全局 hook 自动修复方法，所述定义一个变量函数，所述变量函数用于记录待 hook 的类的名称包括：

获取预置的 key 值；

— 15 —

根据所述预置的 key 值调用存储于系统存储器或者预置数据库中的崩溃类的数组；

根据所述崩溃类的数组定义一个变量函数，所述变量函数用于记录待 hook 的类的名称。

6、根据权利要求 1 所述的全局 hook 自动修复方法，所述根据所述循环和所述变量函数判断所述待 hook 的类的名称是否在过滤名单中包括：

确定所述循环中调用的数组；

在所述调用的数组中确定所述变量函数；

确定所述变量函数中待 hook 的类的名称，并判断所述待 hook 的类的名称是否在过滤名单中。

7、根据权利要求 1-6 中任一所述的全局 hook 自动修复方法，所述方法还包括：

若所述过滤名单中崩溃类的数量大于阈值，则对引起崩溃的类进行整合；

确定所述崩溃类中相同类型的崩溃类的前缀；

将具有所述前缀的崩溃类进行批量屏蔽。

8、一种全局 hook 自动修复装置，包括：

第一定义单元，用于定义一个循环，所述循环用于对所有的类进行 hook；

第二定义单元，用于定义一个变量函数，所述变量函数用于记录待 hook 的类的名称；

判断单元，用于根据所述循环和所述变量函数判断所述待 hook 的类的名称是否在过滤名单中；

第一屏蔽单元，若所述待 hook 的类的名称在所述过滤名单中，则用于屏蔽与所述待 hook 的类的名称对应的崩溃类，所述崩溃类为引起崩溃或异常的类。

9、根据权利要求 8 所述的全局 hook 自动修复装置，全局 hook 自动修复装置还包括：

调用单元，若所述待 hook 的类的名称不在所述过滤名单中，则用于调用所述循环对所述待 hook 的类进行 hook。

10、根据权利要求 9 所述的全局 hook 自动修复装置，全局 hook 自动修复装置还包括：

监听单元，用于监听异常函数，所述异常函数用于监控并记录所述崩溃类对应的数组；

-16-

第一确定单元，用于根据所述崩溃类对应的数组确定所述异常函数中的崩溃类的名称；
导入单元，用于将所述崩溃类的名称导入到所述过滤名单。

11、根据权利要求 8 所述的全局 hook 自动修复装置，第一定义单元具体用于：

5 获取当前运行的目标应用程序的路径；

获取所述目标应用程序的各个类的名称以及数量；

根据所述各个类的名称以及数量定义一个 for 循环，所述 for 循环用于按照所述目标应用程序的路径对各个类进行全局 hook。

10 12、根据权利要求 1 所述的全局 hook 自动修复装置，第二定义单元具体用于：

获取预置的 key 值；

根据所述预置的 key 值调用存储于系统存储器或者预置数据库中的崩溃类的数组；

根据所述崩溃类的数组定义一个变量函数，所述变量函数用于记录待 hook 的类的名称。

15 13、根据权利要求 8 所述的全局 hook 自动修复装置，判断单元具体用于：

确定所述循环中调用的数组；

在所述调用的数组中确定所述变量函数；

确定所述变量函数中待 hook 的类的名称，并判断所述待 hook 的类的名称是否在过滤名单中。

20

14、根据权利要求 8-13 中任一所述的全局 hook 自动修复装置，全局 hook 自动修复装置还包括：

整合单元，若所述过滤名单中崩溃类的数量大于阈值，则用于对引起崩溃的类进行整合；

25 第二确定单元，用于确定所述崩溃类中相同类型的崩溃类的前缀；

第二屏蔽单元，用于将具有所述前缀的崩溃类进行批量屏蔽。

15、一种全局 hook 自动修复设备，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计算机程序，所述处理器执行所述计算机程序时实现如下步骤：

- 17 -

定义一个循环，所述循环用于对所有的类进行 hook；

定义一个变量函数，所述变量函数用于记录待 hook 的类的名称；

根据所述循环和所述变量函数判断所述待 hook 的类的名称是否在过滤名单中；

5 若所述待 hook 的类的名称在所述过滤名单中，则屏蔽与所述待 hook 的类的名称对应的崩溃类，所述崩溃类为引起崩溃或异常的类。

16、根据权利要求 15 所述的全局 hook 自动修复设备，所述处理器执行所述计算机程序实现所述根据所述循环和所述变量函数判断所述待 hook 的类的名称是否在过滤名单中之后，还包括以下步骤：

10 若所述待 hook 的类的名称不在所述过滤名单中，则调用所述循环对所述待 hook 的类进行 hook。

17、根据权利要求 16 所述的全局 hook 自动修复设备，所述处理器执行所述计算机程序实现所述屏蔽与所述待 hook 的类的名称对应的崩溃类，所述崩溃类为引起崩溃或异常的类型之后，还包括以下步骤：

监听异常函数，所述异常函数用于监控并记录所述崩溃类对应的数组；

根据所述崩溃类对应的数组确定所述异常函数中的崩溃类的名称；

将所述崩溃类的名称导入到所述过滤名单。

20 18、根据权利要求 17 所述的全局 hook 自动修复设备，所述处理器执行所述计算机程序实现所述定义一个循环，所述循环用于对所有的类进行 hook 时，包括以下步骤：

获取当前运行的目标应用程序的路径；

获取所述目标应用程序的各个类的名称以及数量；

25 根据所述各个类的名称以及数量定义一个 for 循环，所述 for 循环用于按照所述目标应用程序的路径对各个类进行全局 hook。

19、根据权利要求 15 所述的全局 hook 自动修复设备，所述处理器执行所述计算机程序实现所述定义一个变量函数，所述变量函数用于记录待 hook 的类的名称时，包括以下步骤：

— 18 —

获取预置的 key 值；

根据所述预置的 key 值调用存储于系统存储器或者预置数据库中的崩溃类的数组；

根据所述崩溃类的数组定义一个变量函数，所述变量函数用于记录待 hook 的类的名称。

5

20、一种计算机可读存储介质，所述计算机可读存储介质中存储有指令，当所述指令在计算机上运行时，使得计算机执行如下步骤：

定义一个循环，所述循环用于对所有的类进行 hook；

定义一个变量函数，所述变量函数用于记录待 hook 的类的名称；

10 根据所述循环和所述变量函数判断所述待 hook 的类的名称是否在过滤名单中；

若所述待 hook 的类的名称在所述过滤名单中，则屏蔽与所述待 hook 的类的名称对应的崩溃类，所述崩溃类为引起崩溃或异常的类。

— 1/4 —

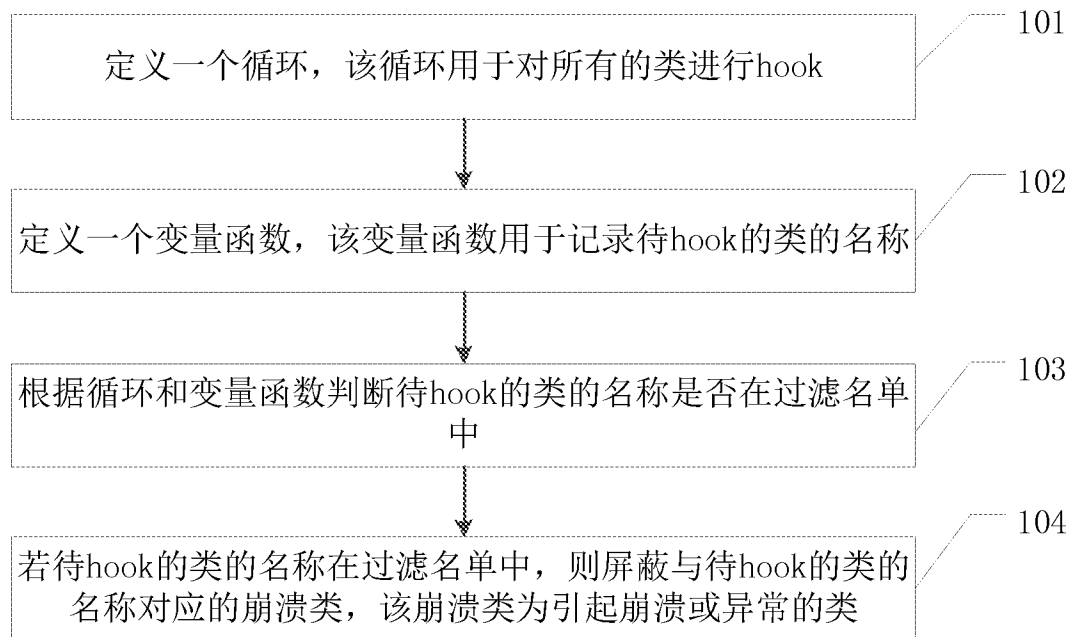


图 1

- 2/4 -

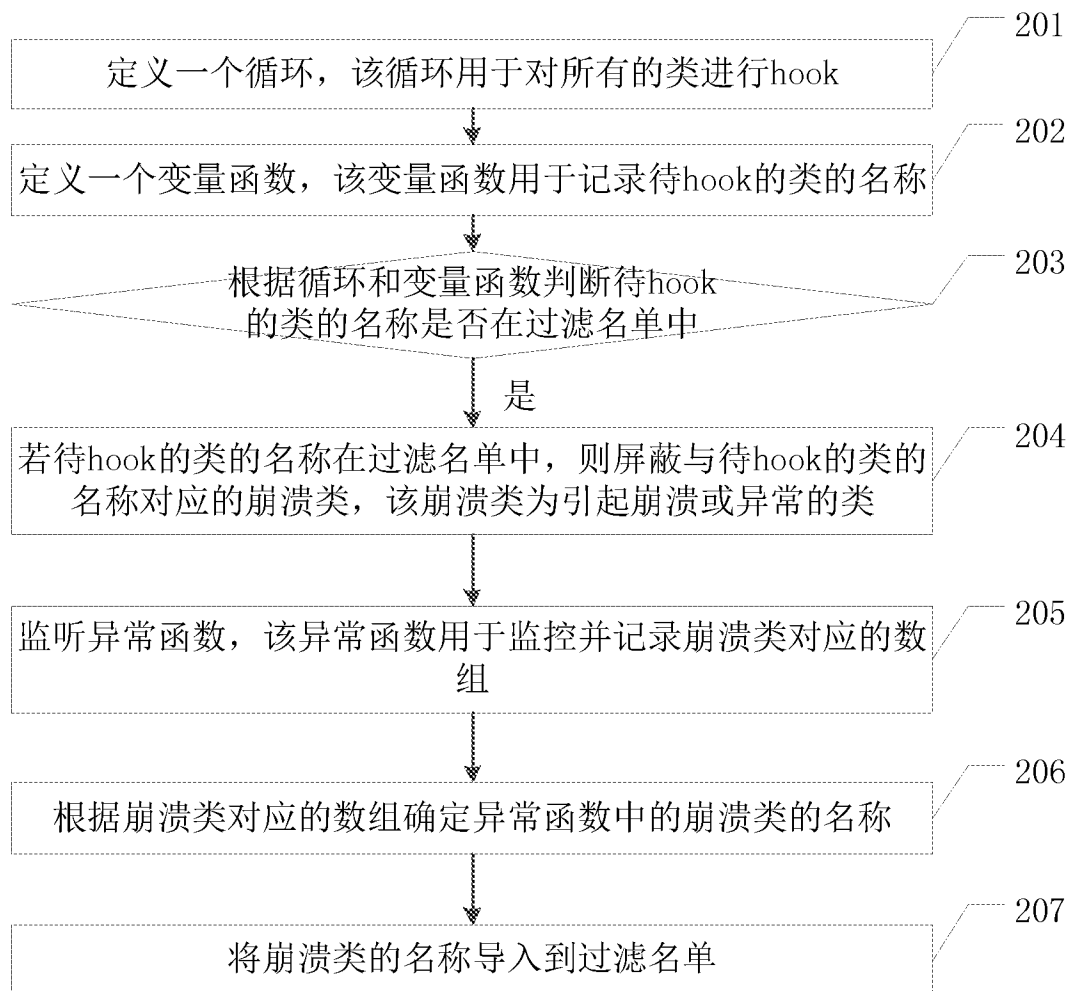


图 2

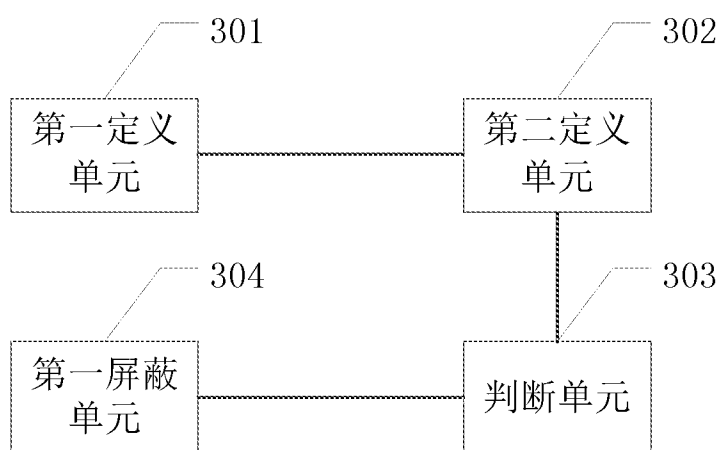


图 3



图 4

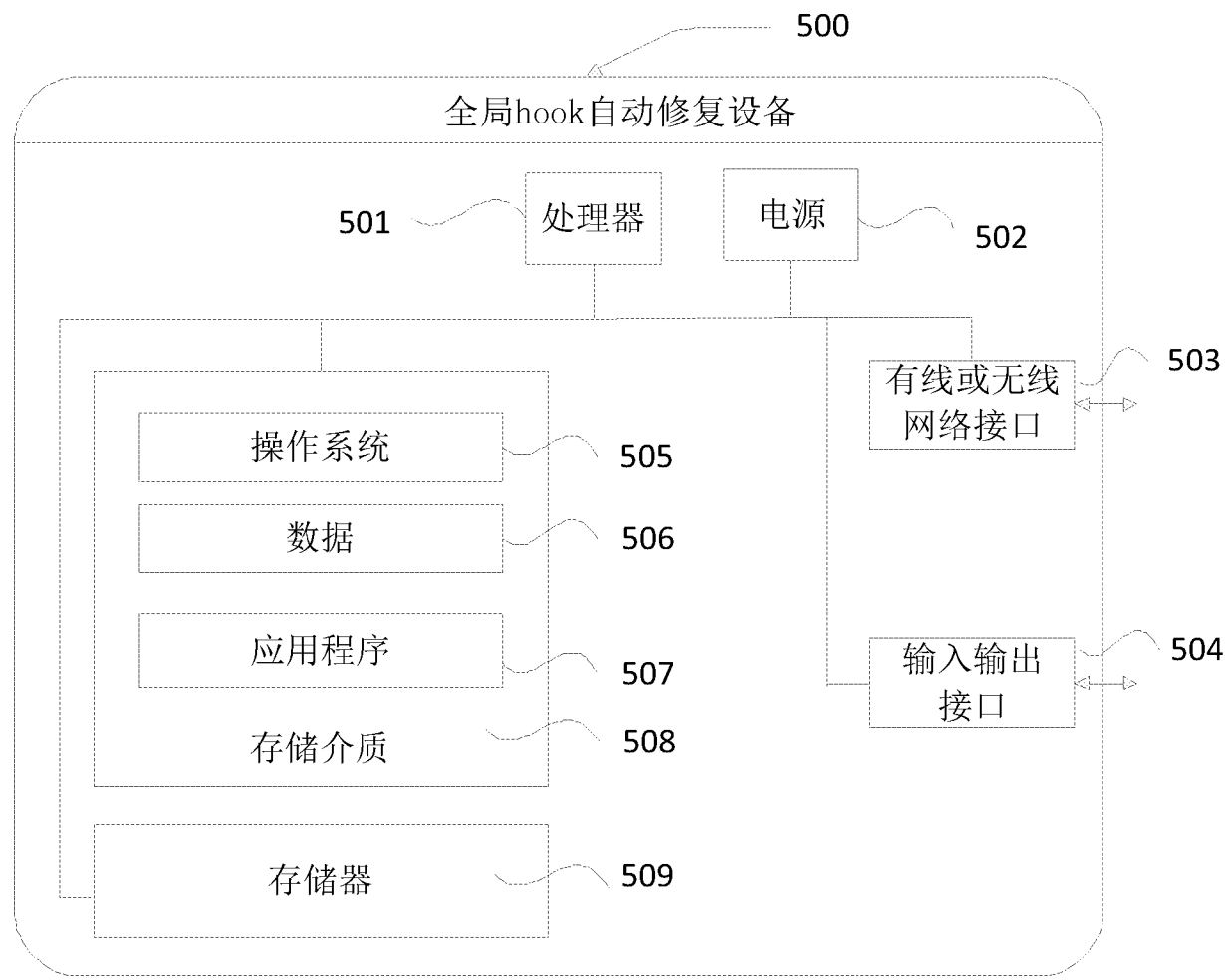


图 5

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2019/103162

A. CLASSIFICATION OF SUBJECT MATTER

G06F 11/36(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)

G06F11/-

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)

USTXT; EPTXT; CNTXT; CNABS; WOTXT; VEN; CNKI: 钩子, 函数, 名称, 循环, 过滤名单, 异常, 屏蔽, 崩溃, 全局, 自动修复, hook, function, name, cycle, filter list, abnormal, shield, breakdown, global, self repair

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CN 106997313 A (TENCENT TECHNOLOGY SHENZHEN CO., LTD.) 01 August 2017 (2017-08-01) description, paragraphs 64-82, 123, and figures 4, 7, 11	1-20
Y	CN 104966018 A (HUAQIAO UNIVERSITY) 07 October 2015 (2015-10-07) description, paragraphs 17-84, and figure 1	1-20
A	CN 104050418 A (ALIBABA GROUP HOLDING LIMITED) 17 September 2014 (2014-09-17) entire document	1-20
A	CN 104992123 A (ZHONG'AN BITE (JIANGSU) SOFTWARE TECHNOLOGY CO., LTD.) 21 October 2015 (2015-10-21) entire document	1-20
A	US 2012304160 A1 (RIDGEWAY INTERNET SECURITY LLC) 29 November 2012 (2012-11-29) 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

23 December 2019

Date of mailing of the international search report

14 January 2020

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing
100088
China

Authorized officer

Facsimile No. (86-10)62019451

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2019/103162

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	106997313	A	01 August 2017	None			
CN	104966018	A	07 October 2015	None			
CN	104050418	A	17 September 2014	CN	104050418	B	13 October 2017
				HK	1201603	A0	04 September 2015
				HK	1201603	A1	27 July 2018
CN	104992123	A	21 October 2015	None			
US	2012304160	A1	29 November 2012	US	9116717	B2	25 August 2015

国际检索报告

国际申请号

PCT/CN2019/103162

A. 主题的分类

G06F 11/36(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)

G06F11/-

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))

USTXT;EPTXT;CNTXT;CNABS;WOTXT;VEN;CNKI:钩子, 函数, 名称, 循环, 过滤名单, 异常, 屏蔽, 崩溃, 全局, 自动修复, hook, function, name, cycle, filter list, abnormal, shield, breakdown, global, self repair

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
Y	CN 106997313 A (腾讯科技深圳有限公司) 2017年 8月 1日 (2017 - 08 - 01) 说明书第64-82、123段及图4、7、11	1-20
Y	CN 104966018 A (华侨大学) 2015年 10月 7日 (2015 - 10 - 07) 说明书第17-84段及图1	1-20
A	CN 104050418 A (阿里巴巴集团控股有限公司) 2014年 9月 17日 (2014 - 09 - 17) 全文	1-20
A	CN 104992123 A (中安比特江苏软件技术有限公司) 2015年 10月 21日 (2015 - 10 - 21) 全文	1-20
A	US 2012304160 A1 (RIDGEWAY INTERNET SECURITY LLC) 2012年 11月 29日 (2012 - 11 - 29) 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

2019年 12月 23日

国际检索报告邮寄日期

2020年 1月 14日

ISA/CN的名称和邮寄地址

中国国家知识产权局(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

授权官员

夏雪

电话号码 86-(20)-28950718

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2019/103162

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	106997313	A	2017年 8月 1日	无			
CN	104966018	A	2015年 10月 7日	无			
CN	104050418	A	2014年 9月 17日	CN	104050418	B	2017年 10月 13日
				HK	1201603	A0	2015年 9月 4日
				HK	1201603	A1	2018年 7月 27日
CN	104992123	A	2015年 10月 21日	无			
US	2012304160	A1	2012年 11月 29日	US	9116717	B2	2015年 8月 25日