



(51) International Patent Classification:
G06F 11/07 (2006.01)

(21) International Application Number:
PCT/EP2023/025441

(22) International Filing Date:
23 October 2023 (23.10.2023)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
202211409719.1 10 November 2022 (10.11.2022) CN

(71) Applicant: **MERCEDES-BENZ GROUP AG** [DE/DE];
Mercedesstraße 120, 70372 Stuttgart (DE).

(72) Inventor: **WANG, Zhe**; Wangjing Street 8, Beijing,
100102 (CN).

(74) Agent: **NOVAGRAAF GROUP**; Chemin de l'Echo 3,
1213 Onex Geneva (CH).

(81) Designated States (*unless otherwise indicated, for every
kind of national protection available*): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ,
CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,

HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG,
KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY,
MA, MD, MG, MK, MN, MU, 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, ST, SV, SY, TH,
TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS,
ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every
kind of regional protection available*): ARIPO (BW, CV,
GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST,
SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ,
RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ,
DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT,
LU, LV, MC, ME, 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).

Published:
— with international search report (Art. 21(3))

(54) Title: METHOD AND SYSTEM FOR HANDLING APPLICATION CRASH, AND DEVICE FOR RUNNING APPLICATION

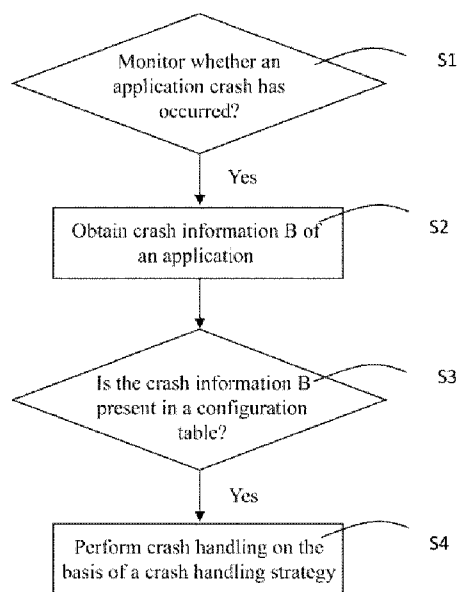


FIG. 1

(57) Abstract: The present invention relates to a method for handling an application crash, comprising: monitoring whether the application crash has occurred; if the crash is monitored, obtaining crash information of the application; and judging whether the crash information of the application is present in a configuration table of application crash handling strategies, the configuration table comprising a stored mapping relationship between crash information of applications and crash handling strategies; and if the crash information of the application is present, performing crash handling on the basis of a crash handling strategy corresponding to the crash information of the application in the configuration table. The present invention further relates to a device for running an application, a system for handling an application crash, and a computer program product. According to the present invention, the crash handling strategy can be automatically matched and executed according to the crash information of the application without user participation, thereby improving the efficiency of the application crash handling and the stability of the application operation.



METHOD AND SYSTEM FOR HANDLING APPLICATION CRASH, AND DEVICE FOR RUNNING APPLICATION

TECHNICAL FIELD

- 5 [0001] The present invention relates to the technical field of computers, and in particular, to a method for handling an application crash, a device for running an application, a system for handling an application crash, and a computer program product for performing steps of the method.

10

BACKGROUND

- [0002] More and more applications (Apps) are developed and deployed on smart terminals (e.g., computers, mobile terminals, in-vehicle infotainment systems of smart networked cars, etc.). For example, with the development of smart networked cars, more and more applications are developed and
15 deployed on a user's mobile terminal and/or vehicle (in-vehicle infotainment) to provide or support various functions or services related to the vehicle. During use of these applications, the applications may crash due to different reasons. This presents challenges to the stability of the applications.

- [0003] Solutions to application crashes proposed in the prior art mainly
20 handle the application crashes by means of a single or fixed manner. For example, after monitoring a program crash, user data is cleared to resume or restart. However, a single or fixed handling mode (e.g., clearing the user data and restarting the application) does not flexibly address different kinds of application crash issues, with great application limitations, especially when the
25 program crash cannot be eliminated after repeatedly clearing user data and restarting the program, or even uninstalling and reinstalling.

- [0004] Therefore, how to flexibly efficiently handling various application crash issues becomes a technical challenge that needs to be solved currently.

30

SUMMARY

- [0005] An objective of the present invention is to provide a method for handling an application crash, a device for running an application, a system for

handling an application crash and a computer program product for performing steps of the method, so as to at least partially address the problem in the prior art. The present invention is based on the following basic concepts: a configuration table of application crash handling strategies is pre-stored in a configuration table stored locally on a client device, and by continuously monitoring the occurrence of application crashes, when an application crash is detected, a crash handling strategy matching the occurred application crash is selected and executed. According to certain embodiments of the present invention, the crash handling strategy can be automatically matched and executed according to the crash information of the application without user participation, thereby improving the efficiency of the application crash handling and the stability of the application operation.

[0006] According to a first aspect of the present invention, there is provided a method for handling an application crash, the method comprising:

15 [0007] monitoring whether an application crash has occurred;

[0008] if it is monitored that the application crash has occurred, obtaining crash information of the application; and

[0009] judging whether the crash information of the application is present in a configuration table of application crash handling strategies, wherein the configuration table comprises a stored mapping relationship between crash information of applications and crash handling strategies; and

20 [0010] if the crash information of the application is present in the configuration table, performing crash handling on the basis of a crash handling strategy corresponding to the crash information of the application in the configuration table.

[0011] According to an optional embodiment of the present invention, in particular, the crash handling may be automatically performed on the basis of the crash handling strategy corresponding to the crash information of the application in the configuration table without being on the basis of user input.

30 [0012] According to an optional embodiment of the present invention, the configuration table of application crash handling strategies may be stored

locally on a client device running the application, and optionally, the configuration table may be remotely updated by a back end server.

[0013] According to an optional embodiment of the present invention, after the application is started and/or when the crash information of the application is not present in the configuration table of application crash handling strategies stored in a client local storage module, it may be queried whether the back end server has an updated configuration table of application crash handling strategies. If it is queried whether the back end server has an updated configuration table of application crash handling strategies, then the updated configuration table may be downloaded from the back end server, and the local configuration table of the client device where the application is located may be updated. In this way, it can be ensured that the latest configuration table participates in the matching, thereby avoiding crash handling strategy matching failure caused by failure to update the configuration table in time.

[0014] According to an optional embodiment of the present invention, the crash information of the application, for example, may comprise type information and/or identification information of an application crash, wherein the type information, for example, comprises a crash type, and the identification information, for example, comprises a full class name, a method name, and/or a file name, and wherein the full class name comprises a class name and/or a packet name.

[0015] According to an optional embodiment of the present invention, the method may further comprise:

[0016] after performing crash handling of the application crash, monitoring whether a preset number of application crashes have occurred over a preset period of time; and

[0017] if the preset number of application crashes have occurred over the preset period of time, stopping performing the crash handling and reporting to a back end server.

[0018] According to an optional embodiment of the present invention, matching of the obtained crash information of the application and the crash information of the application in the configuration table may be performed by

means of fuzzy judgment, wherein a judgment basis parameter on which the fuzzy judgment is based can be dynamically adjusted, especially added and/or removed, according to the obtained crash information of the application and/or the crash information of the application in the configuration table, thereby
5 improving the flexibility and reliability of the matching.

[0019] According to an optional embodiment of the present invention, the crash handling strategy may comprise, for example, one or more of the following strategies:

[0020] doing no operation;

10 [0021] displaying an error page and directing a user to clear user data;

[0022] displaying an upgrade page and directing the user to upgrade the application;

[0023] displaying a page to display a recommended solution;

[0024] performing automatic restart;

15 [0025] performing automatic restart and clearing local user data;

[0026] displaying a page to display an application crash report;

[0027] exiting and re-logging in the application;

[0028] re-initializing a software development kit without restarting the application;

20 [0029] a first downgrade strategy and a first upgrade strategy corresponding to the first downgrade strategy; and/or

[0030] a second downgrade strategy and a second upgrade strategy corresponding to the second downgrade strategy.

[0031] According to an optional embodiment of the present invention, a
25 crashed sub-function of the application may be determined on the basis of the crash information of the application, and the crashed sub-function of the application may be turned off on the basis of the first downgrade strategy until the application is restarted and/or execution of the first downgrade strategy is stopped, and the off sub-function is turned on again on the basis of the first
30 upgrade strategy. The application is downgraded by closing the crashed sub-function of the application, so that the user can continue to use other sub-functions that have not crashed.

[0032] According to an optional embodiment of the present invention, it is judged whether the number of times the same type of application crash occurs over a preset period of time exceeds a preset threshold; if the number of times that the same type of application crash occurs over the preset period of time exceeds the preset threshold, it is judged, on the basis of the crash information of the application, whether the determined sub-function belongs to a pre-given sub-function; if the determined sub-function belongs to the pre-given sub-function, the crashed sub-function of the application is turned off on the basis of the second downgrade strategy until the application is upgraded or a back end service provided by a back end server is updated, and the off sub-function is turned on again on the basis of the second upgrade strategy. Here, each sub-function of the application is graded, and an important sub-function is turned off upon monitoring the crash of the important sub-function until the application is upgraded or the back end service provided by the back end server is updated, and the user can continue to use other important sub-functions that have not crashed.

[0033] According to an optional embodiment of the present invention, the method may further comprise: obtaining, on the basis of a parameter of a client, a configuration table of application crash handling strategies adapted to the parameter from a back end server, and synchronously updating a configuration table stored locally in the client, wherein the parameter of the client comprises, for example, an operating system type, an operating system version, an application version, and/or a device model of the client, etc.

[0034] According to an optional embodiment of the present invention, the method may further comprise:

[0035] judging, on the basis of the monitored thread information crash of the application crash, whether the application crash is a main thread crash or a sub-thread crash;

[0036] if the application crash is the sub-thread crash, recording crash information of the sub-thread; and

[0037] if the application is a main thread crash, performing the crash handling of the main thread crash.

[0038] According to a second aspect of the present invention, there is provided a device for running an application. The device may comprise the following components:

[0039] an application crash monitoring module configured to monitor the occurrence of an application crash, and to obtain crash information of the application;

[0040] an application crash protection management module configured to enable and/or deactivate a corresponding application crash handling function on the basis of the crash information of the application and a configuration table of application crash handling strategies, and to enable and/or deactivate a remote update function of the configuration table;

[0041] a crash handling module configured to handle the application crash on the basis of a crash handling strategy corresponding to the obtained crash information of the application; and

[0042] a client local storage module configured to locally store the configuration table, the configuration table comprising a mapping relationship between crash information of applications and crash handling strategies.

[0043] According to an optional embodiment of the present invention, the crash handling module may be configured to automatically perform the crash handling on the basis of the crash handling strategy corresponding to the crash information of the application in the configuration table without being on the basis of user input.

[0044] According to an optional embodiment of the present invention, the configuration table of application crash handling strategies may be stored in the client local storage module, and optionally, the stored configuration table may be remotely updated by a back end server.

[0045] According to an optional embodiment of the present invention, the application crash protection management module may be configured to: after the application is started and/or when the crash information of the application is not present in the configuration table of application crash handling strategies stored in the client local storage module, query whether the back end server has an updated configuration table of application crash handling strategies; if

it is queried that the back end server has the updated configuration table of application crash handling strategies, download the updated configuration table from the back end server and update the configuration table stored in the client local storage module.

- 5 [0046] According to an optional embodiment of the present invention, the crash information of the application may comprise type information and/or identification information of an application crash, wherein the type information, for example, comprises a crash type, and the identification information comprises a full class name, a method name, and/or a file name, and wherein
10 the full class name comprises a class name and/or a packet name.

[0047] According to an optional embodiment of the present invention, the application crash protection management module may be configured to: monitor whether a preset number of application crashes have occurred over a preset period of time, and stop performing the crash handling and report to the
15 back end server when the preset number of application crashes have occurred over the preset period of time.

[0048] According to an optional embodiment of the present invention, the application crash protection management module may be configured to perform matching of the obtained crash information of the application and the
20 crash information of the application in the configuration table by means of fuzzy judgment, wherein a judgment basis parameter on which the fuzzy judgment is based can be dynamically adjusted, especially added and/or removed, according to the obtained crash information of the application and/or the crash information of the application in the configuration table, thereby improving the
25 flexibility and reliability of the matching.

[0049] According to an optional embodiment of the present invention, the crash handling strategy may comprise one or more of the following strategies:

- [0050] doing no operation;
[0051] displaying an error page and directing a user to clear user data;
30 [0052] displaying an upgrade page and directing the user to upgrade the application;
[0053] displaying a page to display a recommended solution;

- [0054] performing automatic restart;
- [0055] performing automatic restart and clearing local user data;
- [0056] displaying a page to display an application crash report;
- [0057] exiting and re-logging in the application;
- 5 [0058] re-initializing a software development kit without restarting the application;
- [0059] a first downgrade strategy and a first upgrade strategy corresponding to the first downgrade strategy; and/or
- [0060] a second downgrade strategy and a second upgrade strategy
- 10 corresponding to the second downgrade strategy.
- [0061] According to an optional embodiment of the present invention, the crash handling module may be configured to: determine a crashed sub-function of the application on the basis of the crash information of the application, and turn off the crashed sub-function of the application on the
- 15 basis of the first downgrade strategy until the application is restarted and/or execution of the first downgrade strategy is stopped, and turn on the off sub-function again on the basis of the first upgrade strategy.
- [0062] According to an optional embodiment of the present invention, the crash handling module may be configured to: judge whether the number of
- 20 times the same type of application crash occurs over a preset period of time exceeds a preset threshold; if the number of times that the same type of application crash occurs over the preset period of time exceeds the preset threshold, judge, on the basis of the crash information of the application, whether the determined sub-function belongs to a pre-given sub-function; if
- 25 the determined sub-function belongs to the pre-given sub-function, turn off the crashed sub-function of the application on the basis of the second downgrade strategy until the application is upgraded or a back end service provided by a back end server is updated, and turn on the off sub-function again on the basis of the second upgrade strategy.
- 30 [0063] According to an optional embodiment of the present invention, the crash handling module may be configured to: obtain, on the basis of a parameter of a client, a configuration table of application crash handling

strategies adapted to the parameter from a back end server, and synchronously update a configuration table stored locally in the client, wherein the parameter of the client comprises, for example, an operating system type, an operating system version, an application version, and/or a device model of the client, etc.

[0064] According to an optional embodiment of the present invention, the application crash monitoring module judges, on the basis of the monitored thread information crash of the application crash, whether the application crash is a main thread crash or a sub-thread crash, and if the application is the main thread crash, the crash handling of the main thread crash is performed.

[0065] According to a third aspect of the present invention, there is provided a system for handling an application crash, the system being used to perform the method according to the present invention. The system may comprise the following components:

[0066] an application crash monitoring module configured to monitor the occurrence of an application crash, and to obtain crash information of the application;

[0067] an application crash protection management module configured to enable and/or deactivate a corresponding application crash handling function on the basis of the crash information of the application and a configuration table of the application crash handling strategies, and to enable and/or deactivate a remote update function of the configuration table;

[0068] a crash handling module configured to handle the application crash on the basis of a crash handling strategy corresponding to the obtained crash information of the application;

[0069] a client local storage module configured to locally store the configuration table at a client running the application, the configuration table comprising a mapping relationship between information of application crashes and downgrade strategies; and

[0070] a back end server configured to remotely update the configuration table in the client local storage module.

[0071] According to a fourth aspect of the present invention, there is provided a computer program product, such as a computer-readable program carrier, comprising computer program instructions that, when executed by a processor, at least assist in implementing the steps of the method according to the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0072] The principles, features, and advantages of the present invention can be better understood via further detailed description of the present invention provided with reference to the accompanying drawings. The accompanying drawings show:

[0073] FIG. 1 shows a flowchart of a method for handling an application crash according to an exemplary embodiment of the present invention;

[0074] FIG. 2 shows a flowchart of a method for handling an application crash according to another exemplary embodiment of the present invention.

[0075] FIG. 3 shows a flowchart of a method for handling an application crash according to another exemplary embodiment of the present invention;

[0076] FIG. 4 shows a flowchart of a method for handling an application crash according to another exemplary embodiment of the present invention;

[0077] FIG. 5 shows a flowchart of a method for handling an application crash according to another exemplary embodiment of the present invention;

[0078] FIG. 6 shows a flowchart of a method for handling an application crash according to another exemplary embodiment of the present invention;

[0079] FIG. 7 shows a flowchart of a method for handling an application crash according to another exemplary embodiment of the present invention;

[0080] FIG. 8 shows a block diagram of a device for running an application according to an exemplary embodiment of the present invention;

[0081] FIG. 9 shows a block diagram of a system for handling an application crash according to an exemplary embodiment of the present invention; and

[0082] FIG. 10 shows a timing diagram for explaining a program crash handling process according to an exemplary embodiment of the present invention.

DETAILED DESCRIPTION

[0083] In order to make the technical problems to be solved by the present invention, the technical solutions, and the beneficial technical effects clearer, the present invention will be described below in further detail with reference to the accompanying drawings and a plurality of exemplary embodiments. It should be understood that specific embodiments described herein are merely used to explain the present invention, but are not used to limit the scope of protection of the present invention.

10 [0084] FIG. 1 shows a flowchart of a method for handling an application crash according to an exemplary embodiment of the present invention.

[0085] The method may include steps S1 to S4. In step S1, it is monitored whether the application crash has occurred. In the present embodiment of the present invention, the application may include a main thread program.

15 Optionally, at least one sub-thread program may also be included. Here, a crash of the main thread program and/or sub-thread program may be monitored by calling the `Thread.setDefaultUncaughtExceptionHandler ()` function, and the crash is captured when the main thread crash and/or sub-thread crash occurs in the application.

20 [0086] If it is monitored that the application crash has occurred, crash information of the application is obtained in step S2. In this main thread program, a polling task may be executed by means of an application crash monitoring module 11, and the polling task is implemented, for example, by calling the `Looper.loop ()` function and may be executed cyclically, thereby
25 continuously monitoring the running state of the main thread program. Upon monitoring the crash of the main thread program, the `try catch ()` function may be called in time, and by means of the function, the crash information of the main thread crash may be captured. Here, the crash information of the application may include, for example, type information of the application crash,
30 the type information including, for example, a crash type. The crash information of the application may further include, for example, identification information of the application crash, the identification information including, for example, a

full class name, a method name and/or a file name, etc. Here, by means of the method name, a program section for performing each sub-function of the application may be indicated. The full class name may include a class name and/or a packet name, and a sub-function affected by the application crash
 5 may be determined by the class name and/or the packet name.

[0087] Illustratively, stack information of the main thread program of the application crash may be extracted by means of an application crash monitoring module 11, and the stack information is parsed into crash
 10 information B of the application by means of a parsing unit 16, the crash information B being passed to an application crash protection management module 12, as shown in the timing diagram of FIG. 10.

[0088] In step S3, it is judged whether the crash information B of the program is present in a configuration table of application crash handling strategies, wherein the configuration table may include a stored mapping relationship
 15 between crash information A of applications and crash handling strategies. Here, the configuration table of application crash handling strategies may be locally stored at a client device running the application. The application crash protection management module 12 may obtain the locally stored configuration table from a client local storage module 14, and parse the configuration table
 20 into the crash information A of the application included in the configuration table by means of the parsing unit 16. By traversing the crash information A of the application in the configuration table, the obtained crash information B of the application is matched with the crash information A of the application in the configuration table, thereby judging whether the obtained crash information B
 25 of the application is present in the configuration table, and the judgment result thereof may be represented by a judgment truth value, for example.

[0089] As illustratively shown in Table 1, the configuration table may include one or more of the following crash handling strategies:

Strategy serial number	Crash handling strategy
1	Doing no operation
2	Displaying an error page and directing a user to clear user data

3	Displaying an upgrade page and directing the user to upgrade the application
4	Displaying a page to display a recommended solution
5	Performing automatic restart
6	Performing automatic restart and clearing local user data
7	Displaying a page to display an application crash report
8	Exiting and re-logging in the application
9	Re-initializing a software development kit without restarting the application
10	First downgrade strategy and First upgrade strategy corresponding to the first downgrade strategy
11	Second downgrade strategy and second upgrade strategy corresponding to the second downgrade strategy

Table 1 shows crash handling strategies included in the configuration table

[0090] In an optional embodiment of the present invention, matching of the obtained crash information B of the application and the crash information A of the application in the configuration table may be performed by means of fuzzy judgment. Here, a matching basis parameter of the fuzzy judgment may include one or more of a crash type, a full class name, a method name, and/or a file name. Program codes for performing the matching process are illustratively shown below. Here, the matching basis parameter includes a crash type (exceptionType), a class name (className), and a file name (fileName). On the basis of the matching basis parameter, the truth value information (isNativeMethod) about whether there is a matching crash handling strategy in the configuration table can be obtained by means of fuzzy judgment, and the matching strategy serial number is obtained when the truth value information is true.

```
[0091]  {"data":[{"exceptionType":"java.lang.RuntimeException",
               "className":"io_realm_internal_OsSharedRealm",
               "fileName":"io_realm_internal_OsSharedRealm.cpp",
               "isNativeMethod": true,
```

```

        "methodName": "nativeGetSharedRealm",
        "downgradeStrategy": 6,
        "matchRules": [ "exceptionType", "fileName",
        "isNativeMethod", "methodName" ] ]}]

```

5 [0092] In addition, the judgment basis parameter can be dynamically adjusted, especially added and/or removed, according to the obtained crash information B of the application and/or the crash information A of the application in the configuration table, thereby improving the flexibility and reliability of the matching. In the matching basis parameter, for example, a
10 method name may be added, and the program section of the crash may be indicated by means of the method name.

[0093] If the crash information B of the application is present in the configuration table (that is, the judgment truth value information is true), the crash handling is performed in step S4 on the basis of a crash handling
15 strategy corresponding to the crash information B of the application in the configuration table. Here, the crash handling may be automatically performed on the basis of the crash handling strategy corresponding to the crash information of the application in the configuration table without being on the basis of user input. In contrast, there is a need in the prior art to notify a user
20 about crash of the application with respect to the application, and automatically select the adapted crash handling strategy by the user, which not only reduces the handling efficiency of the application crash, but also requires the user to have certain handling experience.

[0094] Here, the majority of the crash handling strategies (e.g., crash
25 handling strategies 2 to 6, 8, and 9, etc.) shown in Table 1 is to handle the application crash by clearing the user data, re-logging in the application, automatically restarting the application, re-initializing the software development kit, etc. These application crashes typically occur in the process of starting an updated application. In such application crashes, the user cannot
30 use any sub-functions in the application, but after executing such a crash handling strategy, the user can use the application normally.

[0095] Illustratively, some sub-functions in the application have exceptions, especially display exceptions, which may be caused by the program codes corresponding to the sub-functions not being executed due to the application crash. If the exceptional sub-function has a smaller impact on the application, for example, the content in a page that is related to the exceptional sub-function displays an exception, the user is not able to see this part of content, whereas other content in the page still displays normal, in particular, other sub-functions of the application are still able to be put into use normally, so that crash handling strategy 1 of doing no operation can be adopted.

10 [0096] For example, a page may also be displayed by means of crash handling strategy 7 to display a crash report, and the crash report may be sent to the development and testing personnel by email for analysis and reporting.

[0097] The crash handling strategy further includes a downgrade strategy and an upgrade strategy corresponding to the downgrade strategy. In the sense of the present invention, "downgrade" may be understood as follows: when an application is exceptional, to control an exceptional impact range, some (e.g., non-important) sub-functions in the application are turned off, thereby ensuring that the important sub-functions of the application operates normally. The sub-functions of the application may be graded on the basis of the priorities and/or dependence of the sub-functions. For example, if one sub-function has a high priority and/or a strong dependency relationship with other sub-functions, the sub-function may be divided into an important sub-function, such as a function of remotely manipulating a vehicle.

25 [0098] In another optional embodiment of the present invention, for the first downgrade strategy and the first upgrade strategy corresponding to the first downgrade strategy, step S4 may include steps S411 to S413, as shown in FIG. 2. In step S411, a crashed sub-function of the application is determined on the basis of the crash information B of the application. Program codes according to another optional embodiment of the present invention are shown below. A program section where a crash occurs in the application can be determined according to the method name (methodName) in the crash information B of the application, and a sub-function affected by the crash can

30

be determined according to the class name (fileName) in the crash information B of the application. Then, a toggle (toggle) name corresponding to the sub-function is issued to the client, and the function can be turned on and/or off by means of the state of the toggle.

5 [0099] {"data":[{"exceptionType":"java.lang.RuntimeException",
"className":"io_realm_internal_OsSharedRealm",
"fileName":"io_realm_internal_OsSharedRealm.cpp",
"isNativeMethod": true,
"methodName":"nativeGetSharedRealm",
10 "downgradeStrategy":10,
"toggle":["ris-mystar-camera"],
"matchRules":["exceptionType", "fileName", "isNativeMethod",
"methodName"]}]}

[0100] In step S412, the crashed sub-function of the application may be
15 turned off on the basis of the first downgrade strategy. In the present embodiment of the present invention, on the basis of the parsed crash information of the application, it can be determined that crash handling strategy 10 is to be adopted, and the toggle state corresponding to the sub-function is set to the off state, so that a user entrance of the sub-function is no longer
20 displayed in the display page of the application, thereby turning off the crashed sub-function of the application, and the user may continue to use other sub-functions that have not crashed.

[0101] Until the application is restarted and/or execution of the first downgrade strategy is stopped, the off sub-function may be turned on again
25 on the basis of the first upgrade strategy corresponding to the first downgrade strategy in step S413, so that all sub-functions of the application recover normal.

[0102] In another optional embodiment of the present invention, for the second downgrade strategy and the second upgrade strategy corresponding
30 to the second downgrade strategy, step S4 may include steps S421 to S425, as shown in FIG. 3. In step S421, it is judged whether the number of times the same type of application crash occurs over a preset period of time exceeds a

preset threshold. Here, on the basis of the crash information B of the application, an application crash type may be determined, and the number of times that the application crash type occurs in the preset period of time can be retrieved. If the application crash type is not retrieved, a crash record regarding the application crash type may be created, so that the crash record can be queried the next time this type of application crash occurs.

[0103] If the number of times the same type of application crash occurs over the preset period of time exceeds the preset threshold, which means that this type of application crash frequently occurs, then in step S422, the crashed sub-function of the application may be determined on the basis of the crash information B of the application. Likewise, the crashed program section in the application may be determined according to the method name in the crash information B of the application, and the sub-function affected by the crash is determined according to the class name and/or packet name in the crash information B of the application. Then, the toggle name corresponding to the sub-function is issued to the client.

[0104] After the sub-function is determined, it is judged in step S423 whether the determined sub-function belongs to a pre-given sub-function, the pre-given sub-function including a function of remotely manipulating a vehicle, for example. If the determined sub-function belongs to the pre-given sub-function, it may be determined that crash handling strategy 11 is to be adopted, then in step S424, the crashed sub-function of the application is turned off on the basis of the second downgrade strategy. Here, the toggle state corresponding to the sub-function may be set to an off state, thereby turning off the crashed important sub-function of the application (e.g., a function of remotely manipulating a vehicle), and the user may continue to use other important sub-functions that have not crashed. While turning off the sub-function, optionally, it is also possible to send to the user prompt information regarding the current sub-function not available by means of a back end server 15.

[0105] Until the application is upgraded or a back end service provided by the back end server 15 is updated, in step S425, the off sub-function is turned on again on the basis of the second upgrade strategy corresponding to the

second downgrade strategy. This is different from executing the first upgrade strategy until the application is restarted and/or execution of the first downgrade strategy is stopped, because this type of application crash occurs frequently. Especially after restarting the application, this type of application crash may continue to occur. Therefore, the second upgrade strategy is executed only after the application has been repaired by means of an application upgrade or a back end service update.

[0106] It can be understood that the crash handling strategy can be adjusted as needed, and is not limited to the above enumerated strategies.

10 [0107] According to the present embodiment of the present invention, the crash handling strategy can be automatically matched and executed according to the crash information of the application without user participation, thereby improving the efficiency of the application crash handling, and ensuring, as much as possible, that sub-functions of the application that have not crashed are still available for use.

[0108] FIG. 4 shows a flowchart of a method for handling an application crash according to another exemplary embodiment of the present invention. Only differences from the embodiment shown in FIG. 1 are described below, and for brevity, the same steps will not be described again.

20 [0109] The method may further include steps S110 and S11. In step S110, it is judged whether the function of application crash handling is turned on. if the function of application crash handling is turned on, the configuration table of application crash handling strategies is remotely updated by the back end server 15 in step S11. In another optional embodiment of the present invention, a configuration table that can be adjusted in real-time according to actual requirements is stored in the back end server 15, and on the basis of a parameter of a client, a configuration table of application crash handling strategies adapted to the parameter may be obtained from the back end server 15, and a configuration table stored locally on the client is synchronously updated, wherein the parameter of the client may include, for example, an operating system type, an operating system version, an application version, and/or a device model of the client, etc. It can be understood that the

configuration table information required for different operating system types, operating system versions, application versions, and/or device models of the client may differ from each other.

[0110] It should be noted that the locally stored configuration table can be
5 synchronously updated to the latest state after starting the application program, and the synchronous update may also be performed after step S3. As shown in FIG. 5, the method may further include steps S10 and S11. If it is judged in step S3 that the crash information B of the application is not present in the configuration table of application crash handling strategies stored in the
10 client local storage module, it is queried in step S10 whether the back end server 15 has an updated configuration table of application crash handling strategies. If it is queried whether the back end server 15 has the updated configuration table of application crash handling strategies, then the updated configuration table is downloaded from the back end server 15 in step S11,
15 and the local configuration table of the client device where the application is located is updated. In this way, it can be ensured that the latest configuration table participates in the matching, thereby avoiding crash handling strategy matching failure caused by failure to update the configuration table in time.

[0111] FIG. 6 shows a flowchart of a method for handling an application
20 crash according to another exemplary embodiment of the present invention. Only differences from the embodiment shown in FIG. 1 are described below, and for brevity, the same steps will not be described again.

[0112] The method may further include steps S5 and S6. In step S5, after performing crash handling of the application crash, it is monitored whether a
25 preset number of application crashes have occurred over a preset period of time. If the preset number of application crashes have occurred over the preset period of time, the crash handling is stopped and reported to the back end server 15 in step S6. In another optional embodiment of the present invention, for example, three application crashes occur over a period of one minute,
30 which means that the application crash is difficult to repair through the crash handling strategy in the current configuration table. Therefore, it may be reported to the back end server 15, so that the development personnel and/or

maintenance personnel of the back end may add to the configuration table a crash handling strategy adapted to the application crash according to the content of the report. In this way, unresolvable application crashes can be reported to the back end personnel in a timely manner, thereby improving the applicability of the crash handling strategy.

[0113] FIG. 7 shows a flowchart of a method for handling an application crash according to another exemplary embodiment of the present invention. Only differences from the embodiment shown in FIG. 1 are described below, and for brevity, the same steps will not be described again.

[0114] The method may further include steps S12 and S13. In step S12, it is judged, on the basis of the monitored thread information crash of the application crash, whether the application crash is a main thread crash or a sub-thread crash. If the application crash is the sub-thread crash, the crash information of the sub-thread crash is recorded in step S13. On the basis of the crash information, buried points and log information may be obtained without crash handling of the sub-thread program. If the application is a main thread crash, it proceeds to step S2 to perform crash handling according to the present invention of the main thread crash. The reliability of the application crash monitoring can be effectively improved by using two or more application crash monitoring modes simultaneously, and the crash occurring in the sub-thread program is recorded.

[0115] In addition, it should be noted that the step number described herein does not necessarily represent an order, but is merely a reference numeral. The order may be changed according to a specific situation, so long as the technical objective of the present invention can be achieved.

[0116] FIG. 8 shows a block diagram of a device for running an application according to an exemplary embodiment of the present invention.

[0117] As shown in FIG. 8, the device 1 may include the following components: an application crash monitoring module 11 configured to monitor the occurrence of an application crash, and to obtain crash information of the application; an application crash protection management module 12 configured to enable and/or deactivate a corresponding application crash

handling function on the basis of the crash information of the application and a configuration table of application crash handling strategies, and to enable and/or deactivate a remote update function of the configuration table; a crash handling module 13 configured to handle the application crash on the basis of

5 a crash handling strategy corresponding to the obtained crash information of the application; and a client local storage module 14 configured to locally store the configuration table, the configuration table including a mapping relationship between crash information A of applications and crash handling strategies. Optionally, the device 1 further includes a parsing unit 16 configured to parse

10 the stack information of the crashed application into crash information B of the application and/or parse the configuration table into crash information A of the application included in the configuration table.

[0118] FIG. 9 shows a block diagram of a system for handling an application crash according to an exemplary embodiment of the present invention. The

15 system 10 may include the device 1 and a back end server 15. The back end server is configured to be able to remotely update the configuration table in the client local storage module. Here, the device 1 may likewise include an application crash monitoring module 11, an application crash protection management module 12, a crash handling module 13, and a client local

20 storage module 14, as shown in the embodiment shown in FIG. 8, which is not described herein again.

[0119] In the following, reference will be made to a timing diagram for explaining an application crash handling process according to an exemplary embodiment of the present invention shown in FIG. 10.

25 [0120] After starting the application, it is monitored in step S1 whether an application crash occurs, and at the same time, it is judged in step S110 whether the function of application crash handling is turned on. If the function of application crash handling is turned on, the application crash protection management module 12 may send an on signal to the application crash

30 monitoring module 11. Upon receiving the on signal, the application crash monitoring module 11 may monitor the occurrence of the application crash, perform step S2 to obtain crash information B of the application, and send the

obtained crash information to the application crash protection management module 12. After the application crash protection management module 12 receives the crash information B of the application, it may be judged in step S3 whether the crash information B of the application is present in the configuration table of application crash handling strategies stored in the client local storage module 14, and step S4 is performed by the crash handling module 13 when the crash information B is present, that is, crash handling is performed on the basis of a crash handling strategy corresponding to the crash information B of the application in the configuration table.

5 [0121] Meanwhile, if the function of application crash handling is judged in step S11 0, then the configuration table of application crash handling strategies is remotely updated by the back end server 15 in step S11. Here, an enable signal of a remote update function may be sent to the application crash protection management module 12, the application crash protection management module 12 obtains, on the basis of a parameter of a client, a configuration table of application crash handling strategies adapted to the parameter from the back end server 15, and synchronously updates the configuration table stored in the client local storage module 14, wherein the parameter of the client includes, for example, an operating system type, an operating system version, an application version, and/or a device model of the client, etc. The parsing unit 16 may parse the updated configuration table into the crash information A of the application included in the configuration table, and send the parsed crash information A to the application crash protection management module 12, so that the crash information B and parsed crash information A of the application may be matched in the application crash protection management module 12.

20 [0122] Although particular embodiments of the present invention are described in detail herein, the particular embodiments are merely for the purpose of explanation, and should not be considered to limit the scope of the present invention. Various replacement schemes and modification schemes may be provided without departing from the core and scope of the present invention.

CLAIMS

1. A method for handling an application crash, the method comprising:
monitoring whether an application crash has occurred;
5 if it is monitored that the application crash has occurred, obtaining crash
information (B) of the application;
judging whether the crash information (B) of the application is present
in a configuration table of application crash handling strategies, wherein the
configuration table comprises a stored mapping relationship between crash
10 information (A) of applications and crash handling strategies; and
if the crash information (B) of the application is present in the
configuration table, performing crash handling on the basis of a crash handling
strategy corresponding to the crash information (B) of the application in the
configuration table.
15
2. The method according to claim 1, wherein crash handling is
automatically performed on the basis of the crash handling strategy
corresponding to the crash information of the application in the configuration
table without being on the basis of user input.
20
3. The method according to claim 1 or 2, wherein the configuration table
of application crash handling strategies is stored locally on a client device
running the application, and the configuration table is remotely updated by a
back end server (15).
25
4. The method according to any one of claims 1 to 3, wherein after the
application is started and/or when the crash information (B) of the application
is not present in the configuration table of application crash handling strategies
stored in a client local storage module, it is queried whether the back end
30 server (15) has an updated configuration table of application crash handling
strategies; and If there is an update, then the updated configuration table is

downloaded from the back end server (15), and the local configuration table of the client device where the application is located is updated.

5. The method according to any one of claims 1 to 4, wherein the crash information (A, B) of the application comprises type information and/or identification information of an application crash, wherein the type information comprises a crash type, and the identification information comprises a full class name, a method name, and/or a file name, and wherein the full class name comprises a class name and/or a packet name.

10

6. The method according to any one of claims 1 to 5, wherein the method further comprises:

after performing crash handling of the application crash, monitoring whether a preset number of application crashes have occurred over a preset period of time; and

15

if the preset number of application crashes have occurred over the preset period of time, stopping performing the crash handling and reporting to the back end server (15).

7. The method according to any one of claims 1 to 6, wherein matching of the obtained crash information (B) of the application and the crash information (A) of the application in the configuration table is performed by means of fuzzy judgment, wherein a judgment basis parameter on which the fuzzy judgment is based can be dynamically adjusted according to the obtained crash information (B) of the application and/or the crash information (A) of the application in the configuration table.

25

8. The method according to any one of claims 1 to 7, wherein the crash handling strategy comprises one or more of the following strategies:

30

doing no operation;

displaying an error page and directing a user to clear user data;

displaying an upgrade page and directing the user to upgrade the application;

displaying a page to display a recommended solution;

performing automatic restart;

5 performing automatic restart and clearing local user data;

displaying a page to display an application crash report;

exiting and re-logging in the application;

re-initializing a software development kit without restarting the application;

10 a first downgrade strategy and a first upgrade strategy corresponding to the first downgrade strategy; and/or

a second downgrade strategy and a second upgrade strategy corresponding to the second downgrade strategy.

15 9. The method according to any one of claims 1 to 8, wherein a crashed sub-function of the application is determined on the basis of the crash information (B) of the application, and the crashed sub-function of the application is turned off on the basis of the first downgrade strategy until the application is restarted and/or execution of the first downgrade strategy is
20 stopped, and the off sub-function is turned on again on the basis of the first upgrade strategy.

10. The method according to any one of claims 1 to 9, wherein it is judged whether the number of times the same type of application crash occurs over a
25 preset period of time exceeds a preset threshold; if the number of times that the same type of application crash occurs over the preset period of time exceeds the preset threshold, it is judged, on the basis of the crash information (B) of the application, whether the determined sub-function belongs to a pre-given sub-function; if the determined sub-function belongs to the pre-given
30 sub-function, the crashed sub-function of the application is turned off on the basis of the second downgrade strategy until the application is upgraded or a

back end service provided by a back end server (15) is updated, and the off sub-function is turned on again on the basis of the second upgrade strategy.

11. The method according to any one of claims 1 to 10, wherein the method
5 further comprises: obtaining, on the basis of a parameter of a client, a configuration table of application crash handling strategies adapted to the parameter from the back end server (15), and synchronously updating a configuration table stored locally in the client, wherein the parameter of the client comprises an operating system type, an operating system version, an
10 application version, and/or a device model of the client.

12. The method according to any one of claims 1 to 11, wherein the method further comprises:
judging, on the basis of the monitored thread information crash of the
15 application crash, whether the application crash is a main thread crash or a sub-thread crash;
if the application crash is the sub-thread crash, recording crash information of the sub-thread; and
if the application is a main thread crash, performing the crash handling
20 of the main thread crash.

13. A device (1) for running an application, characterized by comprising the following components:
an application crash monitoring module (11) configured to monitor the
25 occurrence of an application crash, and to obtain crash information of an application;
an application crash protection management module (12) configured to enable and/or deactivate a corresponding application crash handling function on the basis of the crash information of the application and a configuration
30 table of application crash handling strategies, and to enable and/or deactivate a remote update function of the configuration table;

a crash handling module (13) configured to handle the application crash on the basis of a crash handling strategy corresponding to the obtained crash information of the application; and

5 a client local storage module (14) configured to locally store the configuration table, the configuration table comprising a mapping relationship between crash information (A) of applications and crash handling strategies.

14. The device (1) according to claim 13, wherein the crash handling module (13) is configured to automatically perform the crash handling on the basis of the crash handling strategy corresponding to the crash information of the application in the configuration table without being on the basis of user input.

15. The device (1) according to claim 13 or 14, wherein the configuration table of application crash handling strategies is stored in the client local storage module (14), and the stored configuration table is remotely updated by a back end server (15).

16. The device (1) according to any one of claims 13 to 15, wherein the application crash protection management module (12) is configured to: after the application is started and/or when the crash information (B) of the application is not present in the configuration table of application crash handling strategies stored in the client local storage module, query whether the back end server (15) has an updated configuration table of application crash handling strategies; if there is an update, download the updated configuration table from the back end server (15) and update the configuration table stored in the client local storage module (14).

17. The device (1) according to any one of claims 13 to 16, wherein the crash information of the application comprises type information and/or identification information of an application crash, wherein the type information comprises a crash type, and the identification information comprises a full class

name, a method name, and/or a file name, and wherein the full class name comprises a class name and/or a packet name.

18. The device (1) according to any one of claims 13 to 17, wherein the application crash protection management module (12) is configured to: monitor whether a preset number of application crashes have occurred over a preset period of time, and stop performing the crash handling and report to the back end server (15) when the preset number of application crashes have occurred over the preset period of time.

10

19. The device (1) according to any one of claims 13 to 18, wherein the application crash protection management module (12) is configured to perform matching of the obtained crash information (B) of the application and the crash information (A) of the application in the configuration table by means of fuzzy judgment, wherein a judgment basis parameter on which the fuzzy judgment is based can be dynamically adjusted according to the obtained crash information (B) of the application and/or the crash information (A) of the application in the configuration table.

20. The device (1) according to any one of claims 13 to 19, wherein the crash handling strategy comprises one or more of the following strategies:

doing no operation;
displaying an error page and directing a user to clear user data;
displaying an upgrade page and directing the user to upgrade the application;
displaying a page to display a recommended solution;
performing automatic restart;
performing automatic restart and clearing local user data;
displaying a page to display an application crash report;
exiting and re-logging in the application;
re-initializing a software development kit without restarting the application;

a first downgrade strategy and a first upgrade strategy corresponding to the first downgrade strategy; and/or

a second downgrade strategy and a second upgrade strategy corresponding to the second downgrade strategy.

5

21. The device (1) according to any one of claims 13 to 20, wherein the crash handling module (13) is configured to: determine a crashed sub-function of the application on the basis of the crash information (B) of the application, and turn off the crashed sub-function of the application on the basis of the first
10 downgrade strategy until the application is restarted and/or execution of the first downgrade strategy is stopped, and turn on the off sub-function again on the basis of the first upgrade strategy.

22. The device (1) according to any one of claims 13 to 21, wherein the
15 crash handling module (13) is configured to: judge whether the number of times the same type of application crash occurs over a preset period of time exceeds a preset threshold; if the number of times that the same type of application crash occurs over the preset period of time exceeds the preset threshold, judge, on the basis of the crash information (B) of the application,
20 whether the determined sub-function belongs to a pre-given sub-function; if the determined sub-function belongs to the pre-given sub-function, turn off the crashed sub-function of the application on the basis of the second downgrade strategy until the application is upgraded or a back end service provided by the back end server (15) is updated, and turn on the off sub-function again on the
25 basis of the second upgrade strategy.

23. The device (1) according to any one of claims 13 to 22, wherein the crash handling module (13) is configured to: obtain, on the basis of a parameter of a client, a configuration table of application crash handling
30 strategies adapted to the parameter from the back end server (15), and synchronously update a configuration table stored locally in the client, wherein

the parameter of the client comprises an operating system type, an operating system version, an application version, and/or a device model of the client.

24. The device (1) according to any one of claims 13 to 23, wherein the application crash monitoring module (11) judges, on the basis of the monitored thread information crash of the application crash, whether the application crash is a main thread crash or a sub-thread crash, and if the application is the main thread crash, the crash handling of the main thread crash is performed.

25. A system (10) for handling an application crash, the system (10) being used to perform the method according to any one of claims 1 to 12, characterized in that the system (100) comprises one or more of the following components:

an application crash monitoring module (11) configured to monitor the occurrence of an application crash, and to obtain crash information (B) of the application;

an application crash protection management module (12) configured to enable and/or deactivate a corresponding application crash handling function on the basis of the crash information (B) of the application and a configuration table of application crash handling strategies, and to enable and/or deactivate a remote update function of the configuration table;

a crash handling module (13) configured to handle the application crash on the basis of a crash handling strategy corresponding to the obtained crash information (B) of the application;

a client local storage module (14) configured to locally store the configuration table, the configuration table comprising a mapping relationship between information (A) of application crashes and downgrade strategies; and/or

the back end server (15) configured to be able to remotely update the configuration table in the client local storage module.

26. A computer program product, such as a computer-readable program carrier, comprising computer program instructions that, when executed by a processor, at least assist in implementing the steps of the method according to any one of claims 1 to 12.

5

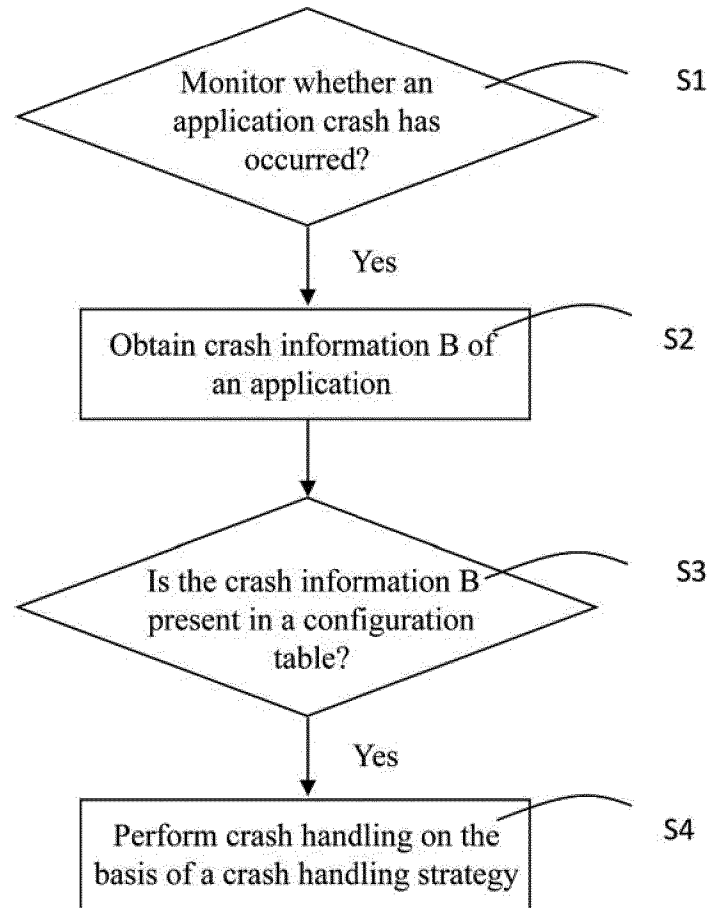


FIG. 1

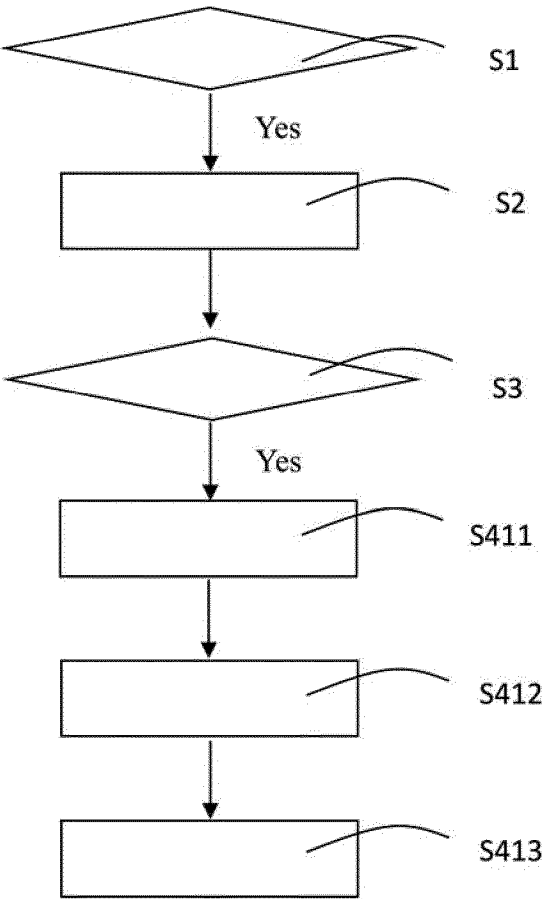


FIG. 2

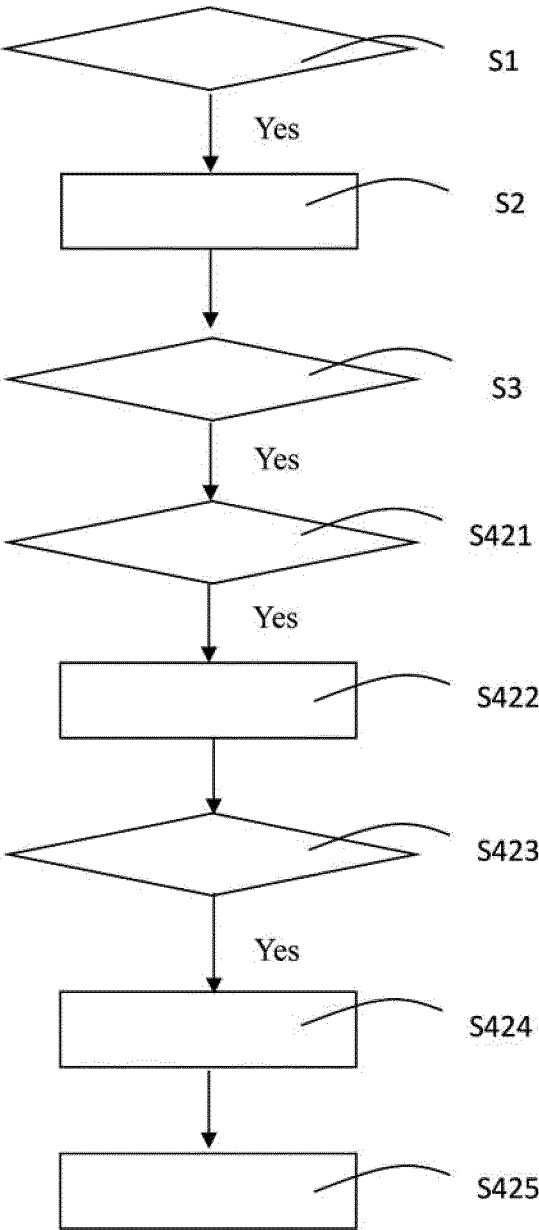


FIG. 3

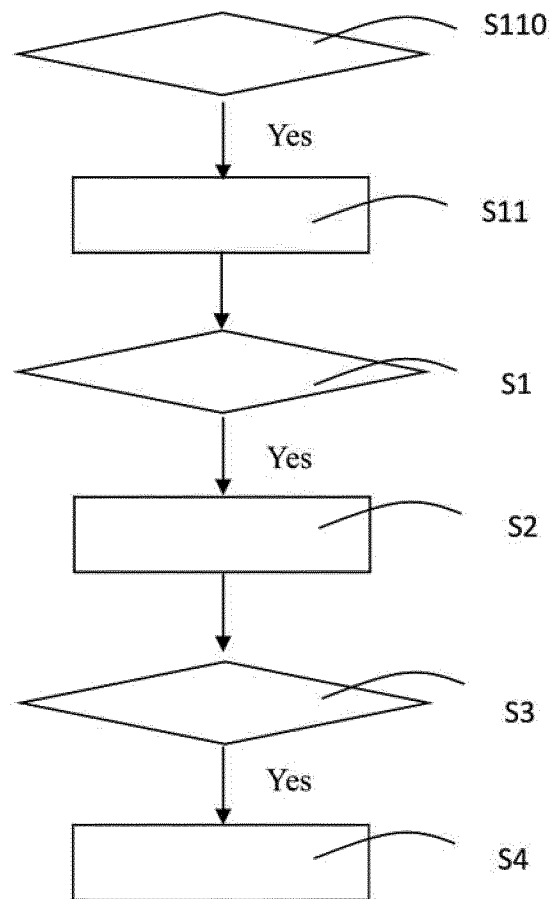


FIG. 4

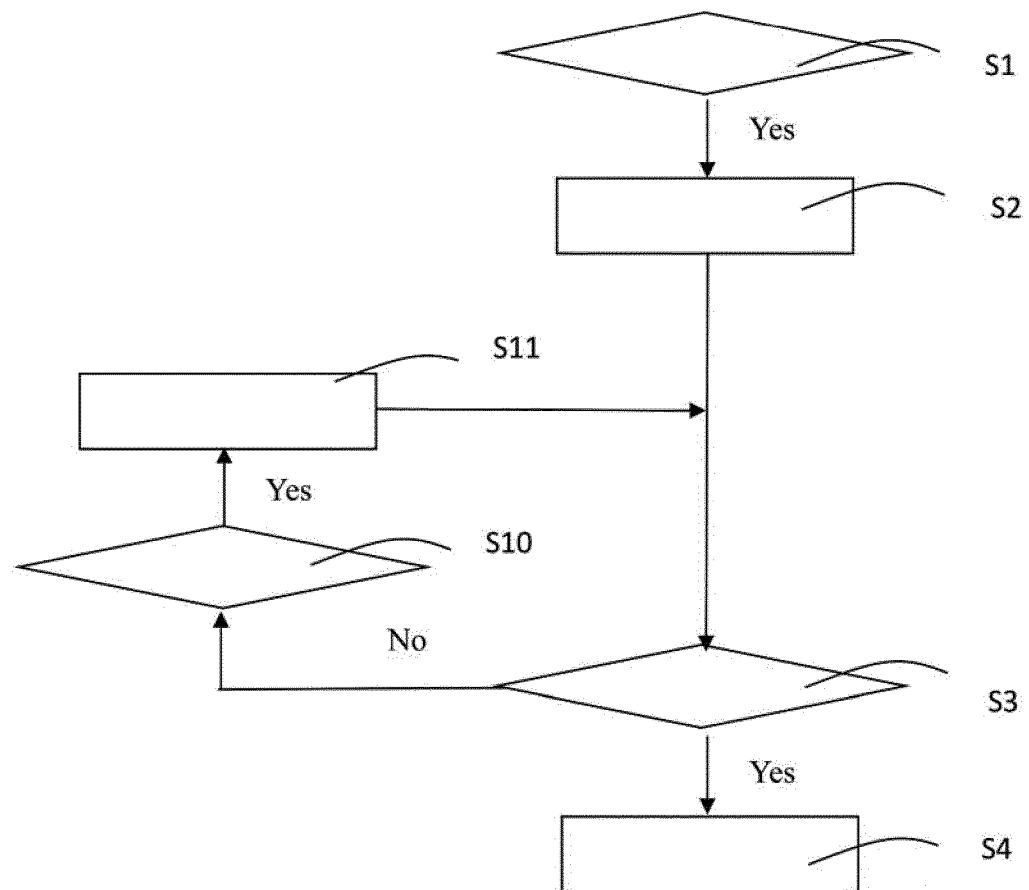


FIG. 5

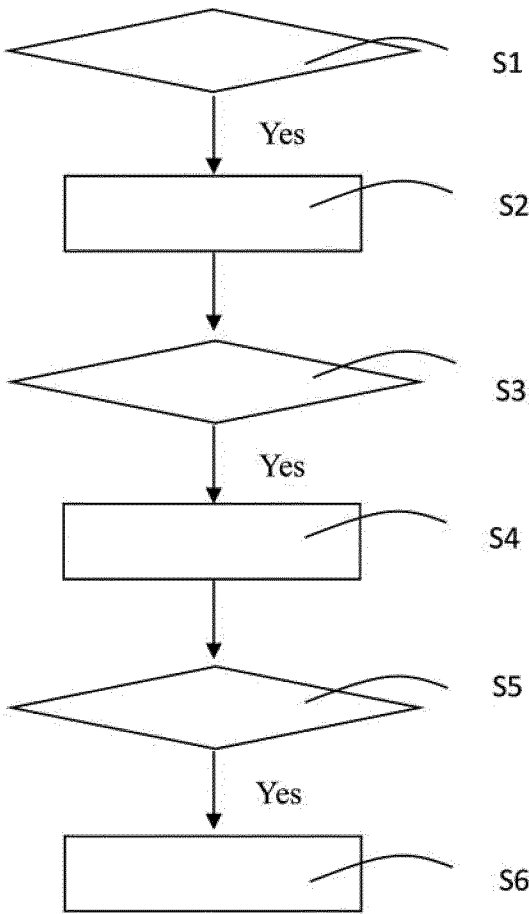


FIG. 6

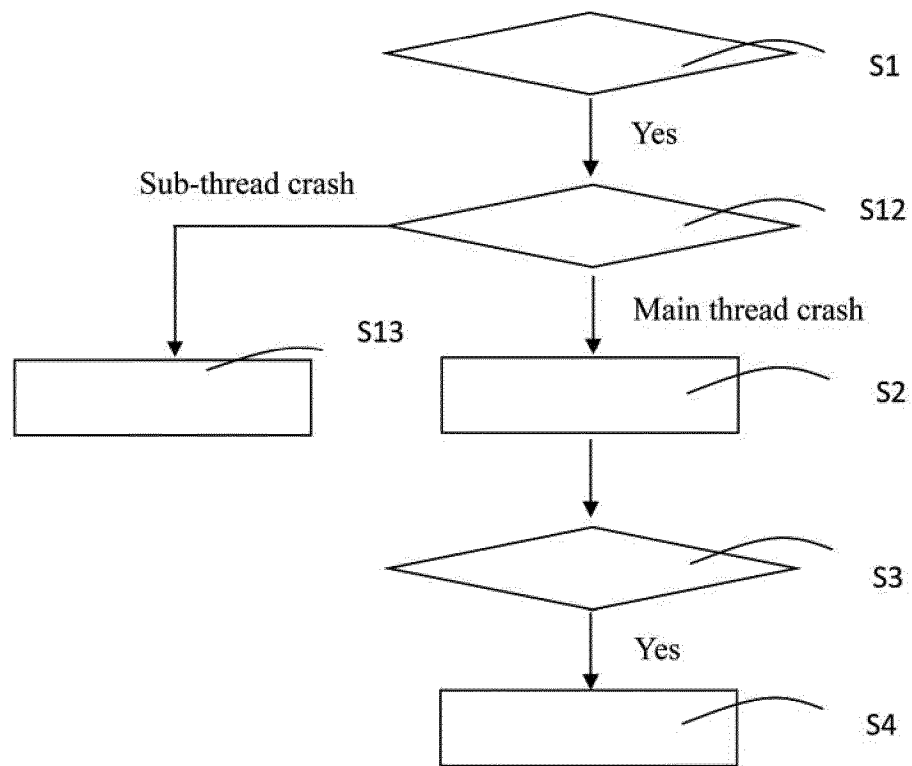


FIG. 7

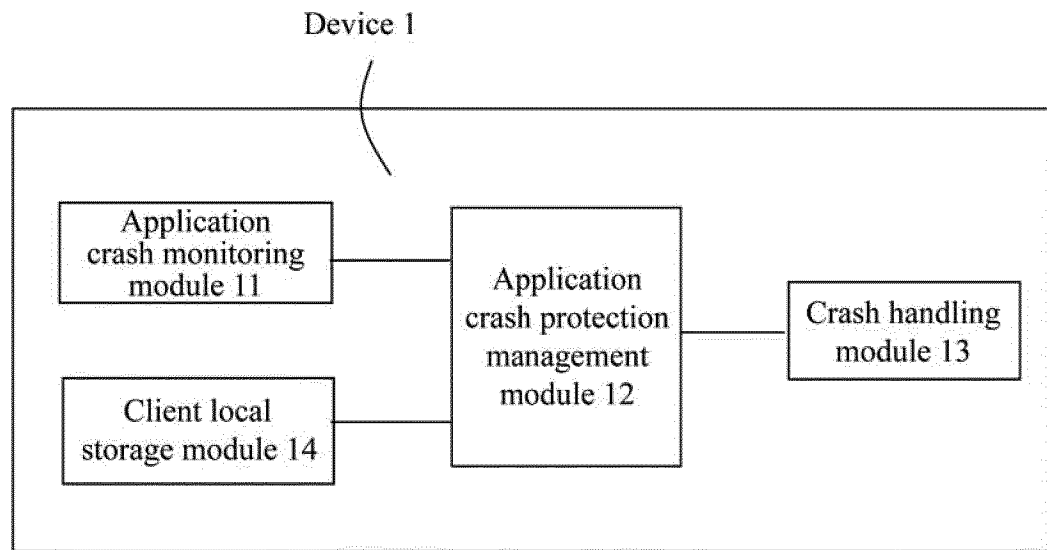


FIG. 8

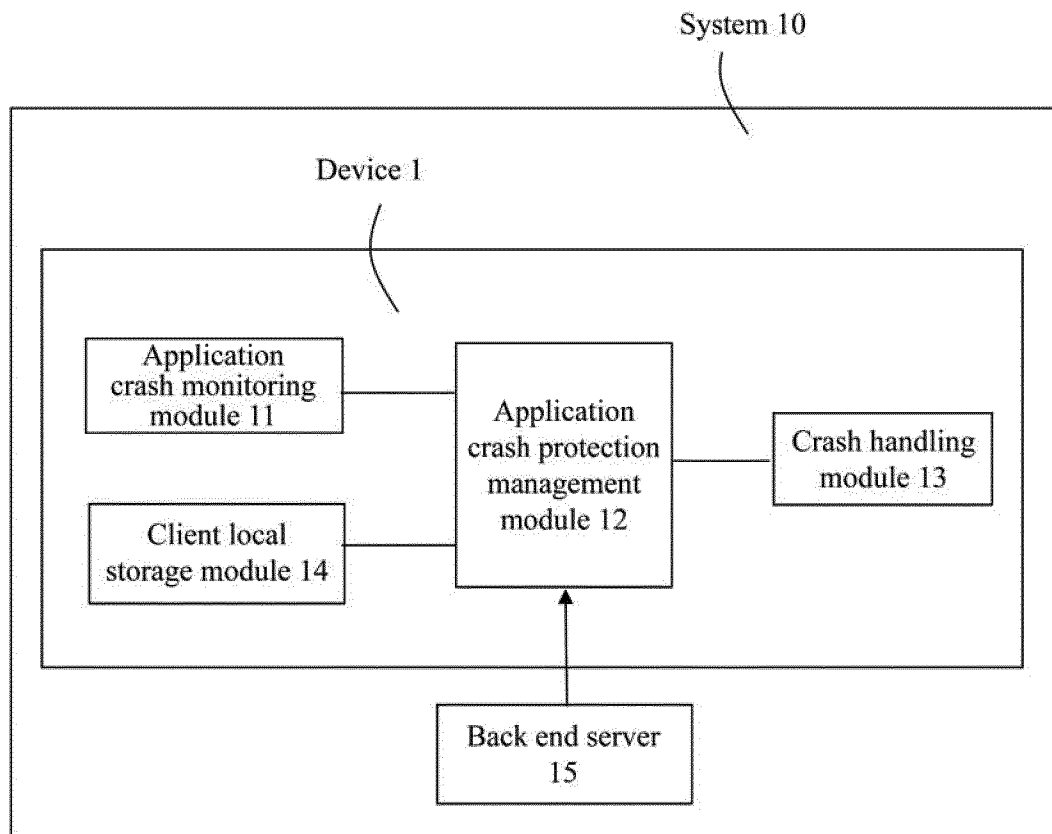


FIG. 9

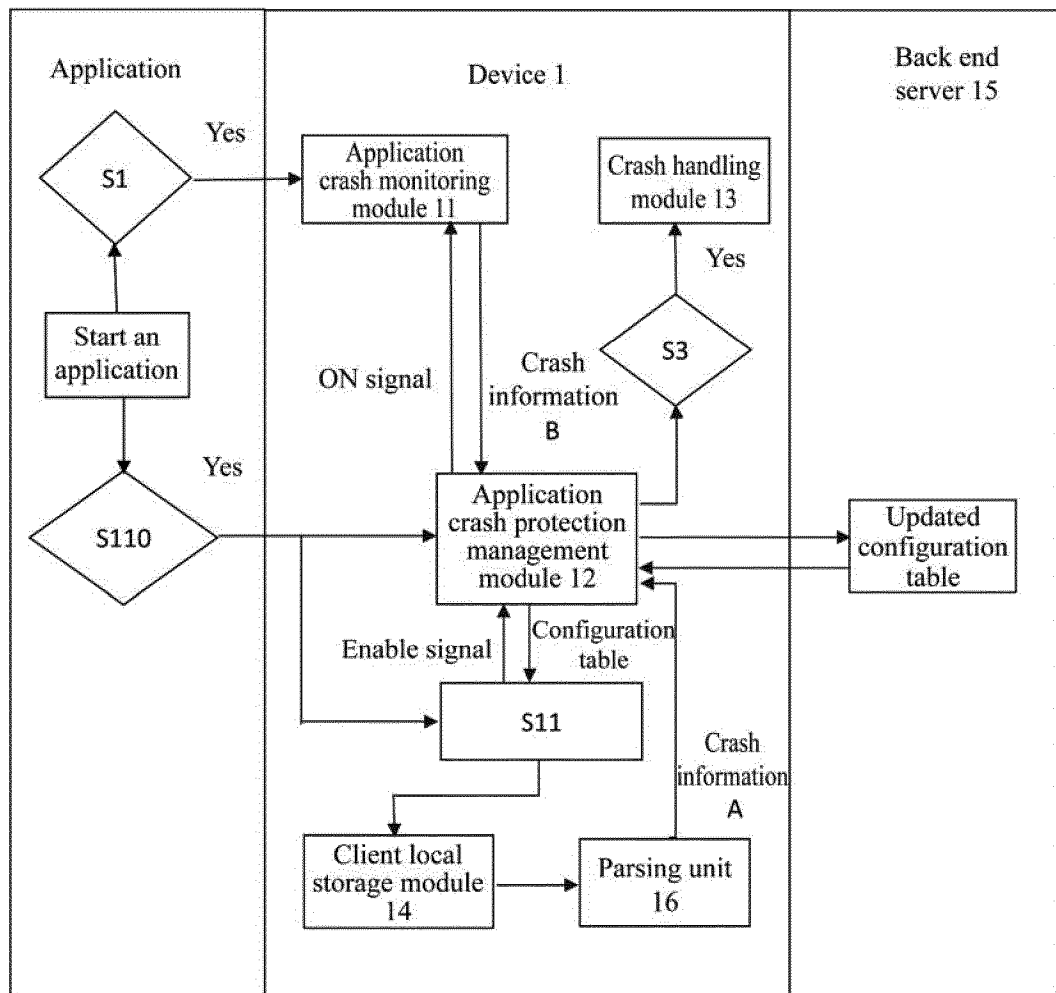


FIG. 10

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2023/025441

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F11/07

ADD.

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)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2021/067607 A1 (GARDNER RICHARD [US] ET AL) 4 March 2021 (2021-03-04) the whole document -----	1-26
X	US 2020/241947 A1 (GONOJI TAKAAKI [JP] ET AL) 30 July 2020 (2020-07-30) the whole document -----	1-26
X	US 2015/271008 A1 (JAIN NAVENDU [US] ET AL) 24 September 2015 (2015-09-24) the whole document -----	1-26



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

12 January 2024

Date of mailing of the international search report

19/01/2024

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2

NL - 2280 HV Rijswijk

Tel. (+31-70) 340-2040,

Fax: (+31-70) 340-3016

Authorized officer

Bauer, Regine

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2023/025441

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2021067607 A1	04-03-2021	NONE	
US 2020241947 A1	30-07-2020	US 2020241947 A1 WO 2019026171 A1	30-07-2020 07-02-2019
US 2015271008 A1	24-09-2015	BR 112016020189 A2 CN 106165345 A EP 3123666 A1 JP 6959736 B2 JP 2017509262 A RU 2016137896 A US 2015271008 A1 US 2019215236 A1 WO 2015148234 A1	15-08-2017 23-11-2016 01-02-2017 05-11-2021 30-03-2017 28-03-2018 24-09-2015 11-07-2019 01-10-2015