

APPLICATION PROGRAM DATA PROCESSING METHOD AND DEVICE

TECHNICAL FIELD

5 [0001] The present application relates to the data processing field, and in particular, to an application program data processing method and device.

BACKGROUND

10 [0002] With the popularization of mobile terminal devices, such as smartphones and tablet computers, people are gradually used to using an application program (APP) to surf the Internet. However, many caches are accumulated after an APP is used for a long time. If the caches are not cleared in time, not only system storage space is occupied, but also the speed of loading the APP is affected.

15 [0003] Currently, a method for clearing an APP cache is mainly as follows: An APP is started, a corresponding function button for clearing the cache is then identified, and tapping the function button can trigger an event of clearing the cache by client software. However, the running speed of the APP is relatively slow when a relatively large amount of caches are accumulated in the APP. Consequently, a relatively long startup time is needed. In addition, after a user taps into the APP, identifying the corresponding function button for clearing the cache also needs search costs, which is time consuming and laborious, and user experience is poor.

20 SUMMARY

[0004] An objective of implementations of the present application is to provide an application program data processing method and device, to quickly and efficiently clear data of an application program without starting the application program, thereby improving user experience.

25 [0005] The present application provides an application program data processing method, where the method includes the following: triggering a pending state of an

application program when detecting a first predetermined user operation in an interface, where the interface has an icon of at least one application program; and in the pending state, processing data of a target application program when detecting a second predetermined user operation on the target application program.

5 [0006] The present application provides an application program data processing device, where the device includes the following: a pending state triggering module, configured to trigger a pending state of an application program when a first predetermined user operation in an interface is detected, where the interface has an icon of at least one application program; and a data processing module, configured to
10 process data of a target application program in the pending state, when a second predetermined user operation on the target application program is detected.

[0007] It can be seen from the technical solutions provided in the implementations described above that, the first predetermined user operation in the interface is detected, and the pending state of the application program is triggered based on the first
15 predetermined operation. Then, in the pending state, an event of processing the data of the target application program can be triggered when the second predetermined user operation on the target application program is detected, so as to complete data processing of the target application program. In an entire process of clearing the data of the application program, the data can be cleared without starting the application
20 program. The data can be quickly and efficiently cleared, and user learning curve can be less steep, so that user experience can be improved.

BRIEF DESCRIPTION OF DRAWINGS

[0008] To describe the technical solutions in the implementations of the present application or in the existing technology more clearly, the following briefly describes
25 the accompanying drawings required for describing the implementations or the existing technology. Apparently, the accompanying drawings in the following description merely show some implementations of the present application, and a person of ordinary skill in the art can still derive other drawings from these accompanying drawings without creative efforts.

30 [0009] FIG. 1 is a flowchart illustrating an application program data processing method, according to an implementation of the present application;

[0010] FIG. 2 is a flowchart illustrating an application program data processing

method, according to an implementation of the present application;

[0011] FIG. 3 is a schematic diagram illustrating an application program data processing method, according to an implementation of the present application;

5 [0012] FIG. 4 is a schematic diagram illustrating an application program data processing method, according to an implementation of the present application;

[0013] FIG. 5 is a flowchart illustrating steps of an application program data processing method, according to an implementation of the present application;

[0014] FIG. 6 is a schematic diagram illustrating an application program data processing device, according to an implementation of the present application;

10 [0015] FIG. 7 is a schematic diagram illustrating an application program data processing device, according to an implementation of the present application;

[0016] FIG. 8 is a schematic diagram illustrating an application program data processing device, according to an implementation of the present application; and

15 [0017] FIG. 9 is a schematic diagram illustrating a data processing module, according to an implementation of the present application.

DESCRIPTION OF IMPLEMENTATIONS

[0018] To make a person skilled in the art understand the technical solutions in the present application better, the following clearly and completely describes the technical solutions in the implementations of the present application with reference to the
20 accompanying drawings in the implementations of the present application. Apparently, the described implementations are merely some but not all of the implementations of the present application. Other implementations obtained by a person of ordinary skill in the art based on the implementations of the present application without creative efforts shall fall within the protection scope of the present application.

25 [0019] The following describes in detail an application program data processing method and device in the present application with reference to the accompanying drawings. FIG. 1 is a flowchart illustrating an application program data processing method, according to an implementation of the present application. Although the present application provides operation steps of a method or a structure of a device
30 shown in the following implementations or the accompanying drawings, the method or the device can include more or fewer operation steps or module structures based on conventional or noncreative efforts. For steps or structures that have no necessary

logical causal relationship, an execution sequence of the steps or a module structure of the device is not limited to an execution sequence or a module structure provided in the implementations of the present application. When the method or the module structure is executed by a device or a terminal product in actual applications, the method or the module structure can be executed based on the sequence of the method or the module structure shown in the implementations or accompanying drawings, or can be executed in parallel (for example, a parallel processor or a multi-thread processing environment).

[0020] Referring to FIG. 1, an application program data processing method according to an implementation of the present application can include the following steps.

[0021] Step S10: Trigger a pending state of an application program when detecting a first predetermined user operation in an interface, where the interface has an icon of at least one application program.

[0022] In the present implementation, the application program data processing method can be applied to a device that has a touchscreen. In addition, another device that completes application control based on interaction with the touchscreen is not excluded. The device described in the present application can include a personal computer (PC), a server computer, a mobile communications terminal, a handheld device, a personal digital assistant (PDA), a tablet computer, a wearable device, an enterprise or public self-service terminal, etc. The device can obtain information about interaction with a subject of a predetermined operation, by using a certain method. The interaction information can include a time, a location, etc. of touching the touchscreen by a subject of the first predetermined operation. When interacting with the user, the device performs data matching by collecting a set of information, such as a time and a location of the user's touch point.

[0023] In the present implementation, an interface is correspondingly disposed on the touchscreen. The interface can be a system interface that has an icon of at least one application program. The application program can be a computer program used to complete one or more tasks. It can: run in a user mode, interact with the user, and has a visual user interface. The application program can include the following: a local service category, such as Dianping and Ding Ding Coupon; an online shopping category, such as Taobao, Jingdong, and Dangdang; a social instant communication tool category, such as WeChat and DingTalk; a financial management category, such

as Ant Fortune and Alipay; and other categories such as an audio or video playback category, a navigation category, and a game category. Certainly, the application program is not limited to the previous classification and enumeration, and is not limited in the present application.

5 **[0024]** In the present implementation, the subject that implements the first predetermined operation can be a hand. In most application scenarios, the user usually uses a finger to tap, press, slide, and shake in a predetermined area in the touchscreen. Certainly, in another implementation, it can be easily figured out that, a corresponding touch control assistant device is used to perform an operation in the predetermined
10 area in the touchscreen. For example, a commonly-used device can include a capacitive stylus, an electromagnetic induction stylus, an infrared-based short-range remote control device, etc. Therefore, the subject that triggers the touchscreen in the present application can include at least one of a hand, a resistive sensing device, a capacitive sensing device, an infrared sensing device, an electromagnetic induction
15 device, an optical sensing device, and a heat sensing device.

[0025] In the present implementation, an electrical signal can be correspondingly generated in the touchscreen based on the first predetermined operation. The electrical signal can be an electrical signal generated by the touchscreen for the subject when the previous enumerated subject is in contact with the touchscreen or is within a
20 sensing distance. In addition, the electrical signal that corresponds to the first predetermined user operation can be superposition of an electrical signal detected by the touchscreen and an electrical signal received by a predetermined function key, or an electrical signal received by a predetermined function key. The application program can be triggered to enter the pending state based on the first predetermined
25 operation.

[0026] In the present implementation, the pending state of the application program can indicate that the application program enters the pending state. In this case, the user can specify a corresponding application program based on an actual need, to perform corresponding processing on the specified application program
30 subsequently. The processing can be processing data of the application program.

[0027] Step S12: In the pending state, process data of a target application program when detecting a second predetermined user operation on the target application program.

[0028] In the present implementation, in the pending state, the data of the

corresponding application program can be processed based on the user's selection. The application program that needs to be cleaned is referred to as a target application program in the following. There may be one or more target application programs, and the number is not limited in the present application. Generally, the running speed of the application program is relatively slow when a relatively large amount of caches are accumulated in the application program. In this case, the user may need to process the data of the application program.

[0029] In an implementation, processing data of a target application program can include at least one of the following: deleting a cache in the target application program; compressing the data in the target application program; and organizing fragments in the target application program. Certainly, the data of the target application program can alternatively be processed by using another method, which is not limited in the present application.

[0030] The cache can be data that is temporarily stored, and usually can be stored in a local disk. The cache can be classified into at least two categories, and the cache includes the following: A first category can be a temporary cache file that is downloaded when the file is opened for viewing, and a second category can be a favorite cache file that the user performs a favorite operation.

[0031] In the present implementation, a subject that implements the second predetermined operation can be a hand. In most application scenarios, the user usually uses a finger to tap, press, slide, and shake in a predetermined area in the touchscreen. Certainly, in another implementation, it can be easily figured out that, a corresponding touch control assistant device is used to perform an operation in the predetermined area in the touchscreen. For example, a commonly-used device can include a capacitive stylus, an electromagnetic induction stylus, an infrared-based short-range remote control device, etc. Therefore, the subject that triggers the touchscreen in the present application can include at least one of a hand, a resistive sensing device, a capacitive sensing device, an infrared sensing device, an electromagnetic induction device, an optical sensing device, and a heat sensing device.

[0032] In the present implementation, an electrical signal can be correspondingly generated in the touchscreen based on the second predetermined operation. The electrical signal can be an electrical signal generated by the touchscreen for the subject when the previous enumerated subject is in contact with the touchscreen or is within a sensing distance. In addition, the electrical signal that corresponds to the

second predetermined user operation can be superposition of an electrical signal detected by the touchscreen and an electrical signal received by a predetermined function key, or an electrical signal received by a predetermined function key. An event of clearing the data of the target application program can be triggered based on the second predetermined operation.

[0033] In the present implementation, in the pending state, the data of the target application program can be processed based on a predetermined rule when the second predetermined user operation on the target application program is detected. The predetermined rule is used to at least maintain a basic function of the target application program. The predetermined rule can be used to ensure that, when the second predetermined operation is received at a coordinate point that corresponds to the target application program, the data of the target application program can be processed, and the target application program itself is not completely or partly deleted.

[0034] The predetermined rule may be as follows: In the pending state, when a sub-icon that can be used to completely or partly delete the target application program appears on an icon that corresponds to the target application program, a location for triggering the target application program to enter a data processing event is set to an icon location of the target program except the sub-icon. In addition, the predetermined rule may be as follows: An original function of the sub-icon is intercepted. When a location of receiving the second predetermined operation is a location where the sub-icon is located, an event of processing the data of the target application program can also be triggered, and the target application program itself is not completely or partly deleted.

[0035] In the present implementation of the present application, the first predetermined user operation in the interface is detected, and the pending state of the application program is triggered based on the first predetermined operation. Then, in the pending state, the event of processing the data of the target application program can be triggered when the second predetermined user operation on the target application program is detected, so as to complete data processing of the target application program. In an entire process of clearing the data of the application program, the data can be cleared without starting the application program. The data can be quickly and efficiently cleared, and user learning curve can be less steep, so that user experience can be improved.

[0036] Referring to FIG. 2, in an implementation, the method can further include the following step:

[0037] Step S14: Display completion information after detecting that data processing of the target application program is completed.

5 **[0038]** In the present implementation, completion of data processing of the target application program may be that, the cache in the target application program is deleted, or the data in the target application program is compressed, or the fragments in the target application program are organized, or may be one or more of the previous cases. Implementations are not limited in the present application. After data
10 processing of the target application program is completed, for example, after it is detected that the cache in the target application program is deleted, or that the data in the target application program is compressed to a predetermined storage capacity, or that the fragments in the target application program are consecutively stored in consecutive disk clusters, or when any combination of the previous cases is detected,
15 the completion information that indicates the completion of data processing of the target application program can be sent to notify the user that data processing of the target application program is completed.

[0039] In an implementation, the completion information can include any one of the following: information for controlling an icon of the target application program to
20 move based on a predetermined track, and information for controlling an icon of the target application program to switch from a first display state to a second display state.

[0040] In the present implementation, after data processing of the target application program is completed, the completion information can be sent to notify
25 the user that data processing of the target application program is completed in this case.

[0041] The completion information can be the information for controlling the icon of the target application program to move based on the predetermined track. For example, when data processing of the target application program is completed, the
30 icon of the target application program can be flipped 360 degrees, or rotated 180 degrees back and forth; or can move based on another track. Implementations are not limited in the present application.

[0042] In addition, when data processing of the target application program is completed, the completion information can be the information for controlling the icon

of the target application program to switch from the first display state to the second display state. For example, the icon of the target application program can be displayed in the first display state when the target application program is in a data clearing process. The first display state can be fuzzy processing, shadow coverage, etc.

5 Certainly, a form of the first display state is not limited to the previous example, and is not limited in the present application. The second display state can be an original icon of the target application program, or a new icon formed by setting a sub-identifier on the basis of the original icon, to indicate that the application program just completes clearing of data in the target application program. Or the second
10 display state can be of another form, which is not limited in the present application.

[0043] Referring to FIG. 3, in an implementation, the method can further include the following step:

[0044] Step S16: Trigger the application program to exit the pending state when detecting a third predetermined user operation in the interface.

15 **[0045]** In the present implementation, after the user clears data of all application programs that needs to be cleaned, the corresponding third predetermined operation can be performed at a predetermined location in the interface, so as to exit the pending state.

[0046] In the present implementation, a subject that implements the third
20 predetermined operation can be a hand. In most application scenarios, the user usually uses a finger to tap, press, slide, and shake in a predetermined area in the touchscreen. Certainly, in another implementation, it can be easily figured out that, a corresponding touch control assistant device is used to perform an operation in the predetermined area in the touchscreen. For example, a commonly-used device can include a
25 capacitive stylus, an electromagnetic induction stylus, an infrared-based short-range remote control device, etc. Therefore, the subject that triggers the touchscreen in the present application can include at least one of a hand, a resistive sensing device, a capacitive sensing device, an infrared sensing device, an electromagnetic induction device, an optical sensing device, and a heat sensing device.

30 **[0047]** In the present implementation, an electrical signal can be correspondingly generated in the touchscreen based on the third predetermined operation. The electrical signal can be an electrical signal generated by the touchscreen for the subject when the previous enumerated subject is in contact with the touchscreen or is within a sensing distance. In addition, the electrical signal that corresponds to the

third predetermined operation can be an electrical signal received by a predetermined function key. Alternatively, the electrical signal that corresponds to the third predetermined user operation can be an electrical signal generated after the user taps a blank location in the interface or presses a predetermined function key, etc. An event
5 that the application program exits the pending state can be triggered based on the third predetermined operation.

[0048] In an implementation, after the step of detecting that data processing of the target application program is completed, the method can further include the following: triggering the application program to exit the pending state.

10 **[0049]** In the present implementation, the application program can directly exit the pending state after step S14 of detecting that data processing of the target application program is completed.

[0050] When there is one target application program that needs to be cleaned, after data processing of the target application program is completed, the application
15 program can exit the pending state to restore to an original state of the interface. When there are a plurality of target application programs that needs to be cleaned, after data processing of the last target application program is completed, the application program can exit the pending state to restore to an original state of the interface.

20 **[0051]** In an implementation, after the step of detecting a second predetermined user operation on the target application program, the method can further include the following: invoking at least two pre-stored deletion options based on the second predetermined operation, where each deletion option corresponds to a cache category in the target application program; and correspondingly performing the step of
25 processing data of the target application program based on a selection result of the user for the deletion options.

[0052] In the present implementation, the cache can be data that is temporarily stored, and usually can be stored in a local disk. The cache can be classified into at least two categories, and the cache includes the following: A first category can be a
30 temporary cache file that is downloaded when the file is opened for viewing, and a second category can be a favorite cache file that the user performs a favorite operation. The deletion options can correspond to the categories of the cache. A first deletion option and a second deletion option can be set for the cache. The first deletion option corresponds to the cache of the first category, and the second deletion option

corresponds to the cache of the second category.

5 [0053] The deletion options can be invoked for the user when the second predetermined user operation is received. When selecting one or more of the deletion options, the user can delete the cache in the corresponding cache category based on the selection result of the user for the deletion options. After the second predetermined operation is received, the deletion options can be presented to the user in a form of a dialog box in the interface. Different deletion options can correspond to different deletion buttons.

10 [0054] In the present implementation, processing the data of the target application program can be deleting the cache in the cache category that corresponds to the target application program. When the user needs to delete a temporary cache file of the first category, the user can choose to tap a corresponding deletion button, so as to delete the cache in the cache category that corresponds to the target application program.

15 [0055] In the present implementation, different deletion options are set for different cache categories, so that the user can selectively delete a cache file that the user does not expect to store, and selectively store information that the user considers valuable, to further improve user experience.

20 [0056] In an implementation, the first predetermined operation can include any one or a combination of the following: a firm press operation, a touch and hold operation, a multi-tap operation, and a shake operation.

25 [0057] In the present implementation, a plurality of coordinate points can be set on the touchscreen. When the first predetermined operation is detected at a coordinate point that is of an icon of the application program and that is correspondingly set in the interface of the touchscreen, a corresponding trigger signal can be generated to trigger the application program to enter the pending state.

30 [0058] In the present implementation, a trigger signal generated based on the firm press operation can be a detected electrical signal of at least one coordinate point in the interface of the touchscreen, that is greater than predetermined pressure. A trigger signal generated based on the touch and hold operation can be an electrical signal that is greater than predetermined duration and that is detected from at least one coordinate point in the interface of the touchscreen, or an electrical signal that is greater than predetermined duration and that is detected at a predetermined function key or function key combination. An input signal generated based on the multi-tap operation can be a plurality of discontinuous electrical signals detected from at least

one coordinate point in the interface of the touchscreen. A time interval between the two adjacent generated electrical signals satisfies predetermined duration. For example, the predetermined duration can be between 1 second and 5 seconds. An input signal generated based on the shake operation can be an input signal generated when an acceleration received by an acceleration sensor reaches a predetermined threshold. A trigger signal generated based on the shake operation can be an acceleration signal generated when the acceleration sensor reaches a specified threshold. The acceleration sensor is integrated into a mobile device, and the acceleration sensor can measure accelerations in three directions: X, Y, and Z. The X, Y, and Z directions are perpendicular to each other, and can form a three-dimensional space coordinate system. A value of the X direction can represent a horizontal movement of the device, a value of the Y direction can represent a vertical movement of the device, and a value of the Z direction can represent a vertical direction of the device in space. When used, the accelerations in the three directions are transmitted to an operating system, and the operating system combines the accelerations in the three directions into a final acceleration value.

[0059] In an example, an electrical signal that corresponds to the first predetermined operation can be an electrical signal generated by touching and holding the icon of the application program in the interface by the user. In addition, the electrical signal that corresponds to the first predetermined operation can be an electrical signal generated by pressing, by the user, a function key disposed on the device or a combination of a plurality of function keys. For example, the first predetermined operation can be pressing a combination of a home key and a volume key to generate an electrical signal. An implementation of the first predetermined operation is not limited to the previous description, and is not listed one by one in the present application. It can be understood that, in the enlightenment of the technical essence of the present application, a person skilled in the art can further make another change on the first predetermined operation, provided that a function implemented and an effect achieved by the first predetermined operation are the same as or similar to those of the present application, and shall be covered within the protection scope of the claims of the present application.

[0060] In an implementation, the second predetermined operation can include any one or a combination of the following: a firm press operation, a touch and hold operation, a multi-tap operation, and a shake operation.

[0061] In the present implementation, a plurality of coordinate points can be set on the touchscreen. When the second predetermined operation is detected at a coordinate point that is of an icon of the application program and that is correspondingly set in the interface of the touchscreen, a corresponding trigger signal can be generated to trigger the application program to enter the pending state.

[0062] In the present implementation, a trigger signal generated based on the firm press operation can be an electrical signal that is greater than predetermined pressure and that is detected from at least one coordinate point in the interface of the touchscreen. A trigger signal generated based on the touch and hold operation can be an electrical signal that is greater than predetermined duration and that is detected from at least one coordinate point in the interface of the touchscreen, or an electrical signal that is greater than predetermined duration and that is detected at a predetermined function key or function key combination. An input signal generated based on the multi-tap operation can be a plurality of discontinuous electrical signals detected from at least one coordinate point in the interface of the touchscreen. A time interval between the two adjacent generated electrical signals satisfies predetermined duration. For example, the predetermined duration can be between 1 second and 5 seconds. An input signal generated based on the shake operation can be an input signal generated when an acceleration received by an acceleration sensor reaches a predetermined threshold. A trigger signal generated based on the shake operation can be an acceleration signal generated when the acceleration sensor reaches a specified threshold. The acceleration sensor is integrated into a mobile device, and the acceleration sensor can measure accelerations in three directions: X, Y, and Z. The X, Y, and Z directions are perpendicular to each other, and can form a three-dimensional space coordinate system. A value of the X direction can represent a horizontal movement of the device, a value of the Y direction can represent a vertical movement of the device, and a value of the Z direction can represent a vertical direction of the device in space. When used, the accelerations in the three directions are transmitted to an operating system, and the operating system combines the accelerations in the three directions into a final acceleration value.

[0063] In an example, an electrical signal that corresponds to the second predetermined operation can be an electrical signal generated by pressing the icon of the application program in the interface and by shaking the device by the user. In addition, the electrical signal that corresponds to the second predetermined operation

can be an electrical signal generated by pressing, by the user, a function key disposed on the device or a combination of a plurality of function keys. An implementation of the second predetermined operation is not limited to the previous description, and is not listed one by one in the present application. It can be understood that, in the enlightenment of the technical essence of the present application, a person skilled in the art can further make another change on the first predetermined operation, provided that a function implemented and an effect achieved by the first predetermined operation are the same as or similar to those of the present application, and shall be covered within the protection scope of the claims of the present application.

10 **[0064]** Referring to FIG. 4, in an implementation, the second predetermined operation is a combination of a shake operation and a press operation, and processing data of a target application program can include the following steps:

[0065] Step S121: Identify the press operation to determine the target application program whose data needs to be processed.

15 **[0066]** Step S123: Process the data of the target application program when detecting that a shake signal reaches a predetermined acceleration threshold.

[0067] In the present implementation, the press operation can be an operation of pressing, by the user in the pending state, the target application program in a predetermined area in the interface of the touchscreen. A trigger signal generated based on the press operation can be an electrical signal that corresponds to a press and that is detected at a predetermined coordinate point in the interface of the touchscreen. In the present implementation, a target application program that needs to be cleaned can be determined by identifying the press operation.

20 **[0068]** In the present implementation, a trigger signal generated based on the shake operation can be an acceleration signal generated when an acceleration sensor reaches a specified threshold. The acceleration sensor is integrated into a mobile device, and the acceleration sensor can measure accelerations in three directions: X, Y, and Z. The X, Y, and Z directions are perpendicular to each other, and can form a three-dimensional space coordinate system. A value of the X direction can represent a horizontal movement of the device, a value of the Y direction can represent a vertical movement of the device, and a value of the Z direction can represent a vertical direction of the device in space. During use, the accelerations in the three directions are transmitted to an operating system, and the operating system combines the accelerations in the three directions into a final acceleration value. Further, the

operating system can compare the final acceleration value with a specified acceleration value, and process the data of the target application program when determining that the shake signal reaches the predetermined acceleration threshold.

[0069] Referring to FIG. 5, in an implementation, in an interface 110, a user can touch and hold an icon of an application program 112. A pending state of the application program is triggered when a device detects the touch and hold operation. In this case, the user can touch and hold a target application program 114 that needs to be cleaned, and shake the device at the same time. When an acceleration signal received by an acceleration sensor in the device reaches a predetermined acceleration threshold, the data of the target application program 114 is cleared. In the clearing process, an icon of the target application program 114 can be rotated 360 degrees. When the clearing succeeds, the icon of the target application program 114 is rotated to the initial location. In this case, it indicates that the data that corresponds to the target application program 114 has been cleared. When data of another application program does not need to be cleared, the user can tap a blank area, press a home key, etc., to exit the pending state, to return to an original state of the interface 110.

[0070] Based on the application program data processing method described in the previous implementations, the present application further provides an application program data processing device.

[0071] Referring to FIG. 6, the application program data processing device can include a pending state triggering module 20 and a data processing module 22.

[0072] The pending state triggering module 20 can be configured to trigger a pending state of an application program when a first predetermined user operation in an interface is detected. The interface has an icon of at least one application program.

[0073] The data processing module 22 is configured to process data of a target application program in the pending state, when a second predetermined user operation on the target application program is detected.

[0074] Referring to FIG. 7, in another implementation of the application program data processing device, the device can further include the following.

[0075] The completion information displaying module 24 can be configured to display completion information after it is detected that data processing of the target application program is completed.

[0076] Referring to FIG. 8, in another implementation of the application program data processing device, the device can further include the following.

[0077] The pending state exit module 26 can be configured to trigger the application program to exit the pending state when a third predetermined user operation in the interface is detected.

5 [0078] Referring to FIG. 9, in another implementation of the application program data processing device, the second predetermined operation is a combination of a shake operation and a press operation, and the data processing module 22 can include an identification unit 221 and a processing unit 223.

10 [0079] The identification unit 221 can be configured to identify the press operation to determine the target application program whose data needs to be processed.

[0080] The processing unit 223 can be configured to process the data of the target application program when it is detected that a shake signal reaches a predetermined acceleration threshold.

15 [0081] The application program data processing device disclosed in the previous implementation corresponds to the implementation of the application program data processing method in the present application, and can implement the implementation of the application program data processing method in the present application and achieve a technical effect of the method implementation.

20 [0082] The above description in the implementations of the present application is merely application of some implementations of the present application, and an implementation slightly modified based on some standards and methods can also be used to implement the solutions in the implementations of the present application. Certainly, other noncreative changes in accordance with steps of the processing method in the implementations of the present application can still be used to
25 implement the same application. Details are omitted here for simplicity.

[0083] Although the present application provides the method steps in the implementations or the flowchart, more or fewer operation steps can be included based on conventional or noncreative efforts. A sequence of the steps enumerated in the implementations is merely one of execution sequences of the steps, and does not
30 represent a unique execution sequence. When the method is executed by a device or a client product in actual applications, the method can be executed based on the sequence of the method in the implementations or accompanying drawings, or can be executed in parallel (for example, a parallel processor or a multi-thread processing environment).

[0084] The devices or modules described in the above implementations can be implemented by a computer chip or an entity, or implemented by a product with a certain function. For ease of description, the previous device is described by dividing the functions into various modules. When the present application is implemented, the functions of all modules can be implemented in one or more pieces of software and/or hardware. Certainly, a module that implements a function can be implemented by using a combination of a plurality of submodules or subunits.

[0085] The method, device, or module described in the present application can be implemented by embedding a readable program in a controller. The controller is implemented based on needs. For example, the controller can use a form such as a microprocessor or a processor and a computer readable medium that stores computer readable program code (e.g., software or firmware) that can be executed by the (micro) processor, a logic gate, a switch, an application specific integrated circuit (ASIC), a programmable logic controller, and an embedded microcontroller. An example of the controller includes but is not limited to the following microcontrollers: ARC 625D, Atmel AT91SAM, Microchip PIC18F26K20, and Silicone Labs C8051F320. The memory and the controller may alternatively be implemented as a part of the control logic of the memory. A person skilled in the art also knows that, in addition to implementing the controller by using the computer readable program code, logic programming can be performed on method steps to allow the controller to implement the same function in forms of the logic gate, the switch, the ASIC, the programmable logic controller, and the embedded microcontroller. Therefore, the controller can be considered as a hardware component, and a device that is included in the controller and configured to implement various functions can also be considered as a structure in the hardware component. Or, the device configured to implement various functions can even be considered as both a software module implementing the method and a structure in the hardware component.

[0086] Some modules in the device in the present application can be described in the general context of an executable computer instruction executed by a computer, for example, a program module. Generally, the program module includes a routine, a program, an object, a component, a data structure, a type, etc. that executes a task or implements an abstract data type. The present application can also be practiced in distributed computing environments. In the distributed computing environments, tasks are performed by remote processing devices that are connected through a

communications network. In a distributed computing environment, the program module can be located in both local and remote computer storage media including storage devices.

[0087] It can be seen from the description of the implementations that, a person skilled in the art can clearly understand that the present application can be implemented by using software in addition to necessary hardware. Based on such an understanding, the technical solutions of the present application essentially, or the part contributing to the existing technology can be implemented in a form of a software product, or can be implemented in a data migration implementation process. The computer software product can be stored in a storage medium, such as a ROM/RAM, a magnetic disk, or an optical disc, and includes several instructions to enable a computer device (which may be a personal computer, a mobile terminal device, a server, a network device, etc.) to execute the method in the implementations or some parts of the implementations of the present application.

[0088] The implementations in the present application are all described in a progressive method. For the same or similar parts in the implementations, reference can be made to these implementations, and each implementation focuses on a difference from other implementations. All or a part of the present application can be used in many general-purpose or dedicated computer system environments or configurations, for example, a personal computer, a server computer, a handheld device or a portable device, a tablet device, a mobile communications terminal, a multiprocessor system, a microprocessor-based system, a programmable electronic device, a network PC, a small computer, a mainframe computer, and a distributed computing environment including any of the above systems or devices.

[0089] In the present application document, a plurality of implementations of the present application are described, and a person skilled in the art can understand that the implementations in the present application can be combined with each other. In addition, a person skilled in the art knows that many modifications and variations of the present application can be made without departing from the spirit of the present application. It is intended that the claims include these modifications and variations without departing from the spirit of the present application.

CLAIMS

What is claimed is:

1. An application program data processing method, comprising:
triggering a pending state of an application program when detecting a first
5 predetermined user operation in an interface, wherein the interface has an icon of at least one application program; and
in the pending state, processing data of a target application program when detecting a second predetermined user operation on the target application program.
2. The method according to claim 1, wherein the method further comprises the
10 following:
displaying completion information after detecting that data processing of the target application program is completed.
3. The method according to claim 2, wherein the method further comprises the following:
15 triggering the application program to exit the pending state when detecting a third predetermined user operation in the interface.
4. The method according to claim 2, wherein after the step of detecting that data processing of the target application program is completed, the method further comprises the following: triggering the application program to exit the pending state.
- 20 5. The method according to claim 1, wherein the processing data of a target application program comprises at least one of the following:
deleting a cache in the target application program;
compressing the data in the target application program; and
organizing fragments in the target application program.
- 25 6. The method according to claim 5, wherein after the step of detecting a second predetermined user operation on the target application program, the method further comprises the following:
invoking at least two pre-stored deletion options based on the second predetermined operation, wherein each deletion option corresponds to a cache
30 category in the target application program; and
correspondingly performing the step of processing data of the target application program based on a selection result of the user for the deletion options.

7. The method according to claim 1, wherein the first predetermined operation comprises any one or a combination of the following:

a firm press operation, a touch and hold operation, a multi-tap operation, and a shake operation.

5 8. The method according to claim 1, wherein the second predetermined operation comprises any one or a combination of the following:

a firm press operation, a touch and hold operation, a multi-tap operation, and a shake operation.

10 9. The method according to claim 1, wherein the second predetermined operation is a combination of a shake operation and a press operation, and the processing data of a target application program comprises the following:

identifying the press operation to determine the target application program whose data needs to be processed; and

15 processing the data of the target application program when detecting that a shake signal reaches a predetermined acceleration threshold.

10. The method according to claim 2, wherein the completion information comprises any one of the following:

information for controlling an icon of the target application program to move based on a predetermined track; and

20 information for controlling an icon of the target application program to switch from a first display state to a second display state.

11. An application program data processing device, comprising:

25 a pending state triggering module, configured to trigger a pending state of an application program when a first predetermined user operation in an interface is detected, wherein the interface has an icon of at least one application program; and

a data processing module, configured to process data of a target application program in the pending state, when a second predetermined user operation on the target application program is detected.

30 12. The device according to claim 11, wherein the device further comprises the following:

a completion information displaying module, configured to display completion information after it is detected that data processing of the target application program is completed.

13. The device according to claim 12, wherein the device further comprises the

following:

a pending state exit module, configured to trigger the application program to exit the pending state when a third predetermined user operation in the interface is detected.

- 5 14. The device according to claim 11, wherein the second predetermined operation is a combination of a shake operation and a press operation, and the data processing module comprises the following:

an identification unit, configured to identify the press operation to determine the target application program whose data needs to be processed; and

- 10 a processing unit, configured to process the data of the target application program when it is detected that a shake signal reaches a predetermined acceleration threshold.