(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 October 2014 (16.10.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/166307 A1

(51) International Patent Classification:
G06F 9/445 (2006.01)

(21) International Application Number:
PCT/CN2014/070763

(22) International Filing Date:
17 January 2014 (17.01.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
201310124995.8 11 April 2013 (11.04.2013) CN

(71) Applicant: **TENCENT TECHNOLOGY (SHENZHEN) COMPANY LIMITED** [CN/CN]; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518000 (CN).

(72) Inventors: **BI, Tingli**; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518000 (CN). **NIE, Kefeng**; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518000 (CN).

(74) Agent: **SHENPAT INTELLECTUAL PROPERTY AGENCY**; Room 1521, West Block, Guomao Building, Shenzhen, Guangdong 518014 (CN).

(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, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, 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, 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 FOR DETERMINING PERFORMANCE OF PROCESS OF APPLICATION AND COMPUTER SYSTEM

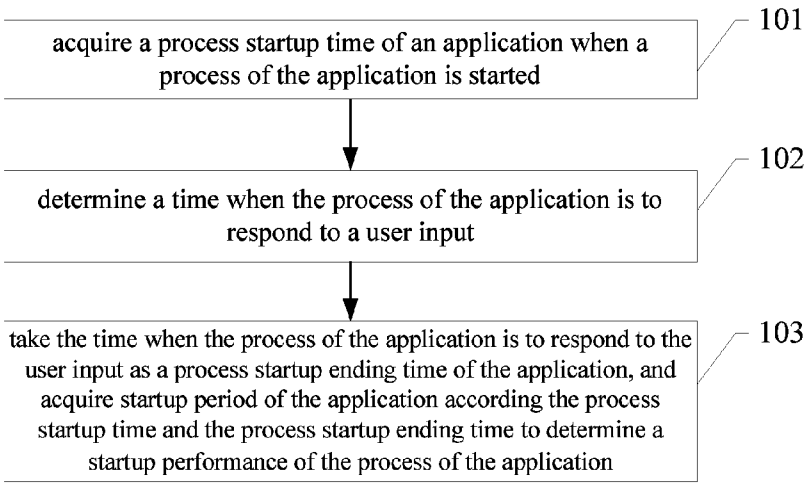

Fig. 1

(57) **Abstract**: A method for determining a performance of a process of an application and a computer system are provided, which are applied in the field of computer technology. In the method, when the process of the application is started, an apparatus for determining a performance of a process of an application acquires a process startup time of the application; determines a time when the process of the application is to respond to a user input; and take the time when the process of the application is to respond to the user input as a process startup ending time of the application and acquire a startup period of the application according to the process startup time and the process startup ending time to determine a startup performance of the process of the application.

WO 2014/166307 A1

METHOD FOR DETERMINING PERFORMANCE OF PROCESS OF APPLICATION AND COMPUTER SYSTEM

[0001] This application claims the priority of Chinese Patent Application No. 201310124995.8, entitled "METHOD FOR DETERMINING PERFORMANCE OF PROCESS OF APPLICATION AND COMPUTER SYSTEM", filed with the Chinese Patent Office on April 11, 2013, which is incorporated by reference in its entirety herein.

FIELD OF THE INVENTION

[0002] The present invention relates to the field of computer technology, and particularly to a method for determining a performance of a process of an application and a computer system.

BACKGROUND OF THE INVENTION

[0003] A performance of a process of an application may be indicated by multiple types of parameters. Specifically, a startup time consumption of the process of the application is an important measurement parameter. If the startup period may be acquired accurately, a startup performance of the process of the application may be indicated accurately.

[0004] In the prior art, the startup period of the process of the application may be acquired by acquiring a process startup ending time t2 and a process startup starting time t1 of the application and calculating a difference between the t2 and the t1. Specifically, a starting time of a user interactive window corresponding to the application may be taken as the process startup ending time t2 of the application. However, processes of some applications, such as a Command Line Interface (CUI) application, do not have user interactive windows. Therefore, the acquired startup period of the process of the application in the prior art may be inaccurate, and thus the startup performance of the process of the application may be determined inaccurately.

SUMMARY OF THE INVENTION

[0005] A method for determining a performance of a process of an application and a

computer system are provided according to embodiments of the disclosure, thus determining the startup performance of the process of the application accurately.

[0006] A method for determining a performance of a process of an application is provided according to an embodiment of the disclosure, which includes:

acquiring a process startup time of the process of the application in the case that the process of the application is started;

determining a time when the process of the application is to respond to a user input as a process startup ending time of the application; and

acquiring startup period of the application according to the process startup time and the process startup ending time to determine a startup performance of the process of the application.

[0007] A computer system is provided according to an embodiment of the present disclosure, which includes:

a process startup unit adapted to start a process of an application;

a startup time acquiring unit adapted to acquire a process startup time of the application in the case that the process of the application is started by the process startup unit;

a response time determining unit adapted to determine a time when the process of the application is to respond to a user input as a process startup ending time of the application; and

a performance determining unit adapted to acquire a startup period of the application according to the process startup time and the process startup ending time to determine a startup performance of the process of the application.

[0008] In the method according to the embodiments of the present disclosure, when a process of an application is started, an apparatus for determining a performance of the process of the application will acquire a process startup time of the application, and determine a time when the process of the application is to respond to a user input, and take the time when the process of the application is to respond to the user input as a process startup ending time of the application and acquire a startup period of the application according to the process startup time and the process startup ending time, so as to determine a startup performance of the

process of the application. Hence, even if processes of some applications do not have user interactive interfaces, there is no influence on the calculating of the startup period of the process of the application. Therefore, the startup period of the process of the application is obtained accurately, thus determining the startup performance of the process of the application accurately.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] In order to more clearly illustrate the technical solution in the embodiments of the present disclosure or in the prior art, in the following, accompanying drawings required in the description of the embodiments or the prior art will be introduced simply. Obviously, the accompanying drawings in the following description are just some embodiments of the present disclosure. Those skilled in the art may also obtain other accompanying drawings according to these accompany drawings without any creative work.

[0010] Figure 1 is a flow chart of a method for determining a performance of a process of an application provided by an embodiment of the present disclosure;

[0011] Figure 2 is a flow chart of another method for determining a performance of a process of an application provided by an embodiment of the present disclosure;

[0012] Figure 3 is a flow chart of another method for determining a performance of a process of an application provided by an embodiment of the present disclosure;

[0013] Figure 4 is a schematic structural diagram of a computer system provided by an embodiment of the present disclosure;

[0014] Figure 5 is a schematic structural diagram of another computer system provided by an embodiment of the present disclosure;

[0015] Figure 6 is a schematic structural diagram of another computer system provided by an embodiment of the present disclosure; and

[0016] Figure 7 is a schematic structural diagram of a terminal to which a method for determining a performance of a process of an application is applied according to an embodiment of the present disclosure.

DETAILED DESCRIPTION OF THE DISCLOSURE

[0017] In the following, the technical solution in the embodiments of the present disclosure will be described clearly and completely in conjunction with the accompanying drawings in the embodiments of the present disclosure. Obviously, the described embodiments are just a part of embodiments of the present disclosure, and are not all embodiments. All other embodiments obtained by those skilled in the art based on the embodiments of the present disclosure without any creative work will fall within the scope of protection of the present disclosure.

[0018] A method for determining a performance of a process of an application is provided according to an embodiment of the present disclosure. The performance of the process of the application is determined mainly by determining a startup period of the process of the application. The method in the embodiment of the present disclosure is performed by an apparatus for determining a performance of a process of an application. A flow chart is shown in Figure 1, and the method includes steps 101-103.

[0019] Step 101, acquiring a process startup time t1 of an application when a process of the application is started.

[0020] It can be understood that the apparatus for determining a performance of a process of an application needs to acquire multiple parameters of the process of the application when determining the startup performance of the process of the application. Specifically, when the startup performance of the process of the application is determined, it is required to firstly acquire a startup period of the process of the application. Thus, the apparatus for determining a performance of a process of an application firstly acquires the process startup time of the application.

[0021] In specific implementation, the apparatus for determining the performance of the process of the application may firstly call an Application Programming Interface (API) of a CreatProcess to create the process of the application. For some processes of applications which cannot be created, the apparatus for determining the performance of the process of the application may prompt a user to start the application manually, such as by clicking an icon of the process of the application. Then, the apparatus for determining the performance of the process of the application may call an API of a GetTickCount to acquire the process startup time of the application.

[0022] Step 102, determining a time when the process of the application is able to respond to a user input as a process startup ending time t2 of the application. Some processes of applications may receive information input by a user and perform a certain operation according to the information input by the user. Herein, the process of the application being able to respond to the user input means that the process of the application is able to receive information input by a user. Specifically, it is determined whether the process of the application may receive a mouse input and/or a keyboard input and so on. That is, it is determined whether the mouse input or the keyboard input may interrupt the process of the application. If the mouse input or the keyboard input may interrupt the process of the application, the process of the application is able to respond to the user input.

[0023] Step 103, acquiring a startup period of the application according the process startup time t1 and the process startup ending time t2 to determine a startup performance of the process of the application. That is, the startup period is a difference between the process startup ending time t2 and the process startup time t1. If the startup period of the application is long, the startup performance of the process of the application is poor, and a further optimization is required.

[0024] It should be illustrated that the process of the application described above refers to an active application. That is, the apparatus for determining the performance of the process of the application has put codes of the application in a storage module thereof, i.e. the process is an application which occupies system resources. An application is referred to "program" before being called into a memory space, and is referred to "process" after being called and having got resources. One process may include many threads. Codes of each application are stored in a space in a corresponding space segment of the storage module, and the process of the application can include multiple execution threads. The step of starting a process of an application includes: loading codes and other information required in the execution of the process of the application into the storage module of the apparatus for determining the performance of the process of the application. In this case, the process startup time of the application is a time for starting the loading of the codes and the other information, and the process startup ending time of the application is a time for completing the loading of the code and other information.

[0025] In the embodiments of the present disclosure, when the process of the application is

started, the apparatus for determining the performance of the process of the application acquires the process startup time of the application, determines the time when the process of the application can respond to the user input, takes the time when the process of the application can respond to the user input as the process startup ending time of the application, and acquires the startup period of the application according to the process startup time and the process startup ending time. In this way, the startup performance of the process of the application is determined. Therefore, even if some processes of application do not have user interactive interfaces, there is no influence on the calculation of the startup period of the process of the application, thus the startup period of the process of the application is obtained accurately, and the startup performance of the process of the application is determined accurately.

[0026] As shown in Figure 2, in an embodiment, in a case of a CUI application, such as an application for realizing a ping command, a mouse input is not supported, and thus an instruction is input by a user via a key board. The process of the application executes the instruction. The step 102 described above may be realized by the apparatus for determining the performance of the process of the application. The step 102 further includes steps A1-A2.

[0027] A1: calling a WaitForIdleInput interface to determine whether the process of the application is to respond to a user input. Specifically, in a case where the apparatus for determining the performance of the process of the application calls the WaitForIdleInput interface, the WaitForIdleInput interface provides a return value when the process of the application has an idle to process the user input; and does not provide the return value when the process of the application has no idle to process the user input.

[0028] A2: taking a time when the WaitForIdleInput interface provides a return value as a time when the process of the application is to respond to the user input.

[0029] Further, when the WaitForIdleInput interface does not provide the return value within a preset first timeout, the apparatus for determining the performance of the process of the application will stop performing the step 102 described above, that is, stop the step of determining a time when the process of the application is to respond to the user input. Hence, in a startup procedure of the process of the application, if the process of the application cannot respond to the user input due to some reasons for example some execution threads in an infinite loop, the apparatus for determining the performance of the process of the application

will stop a procedure of acquiring the startup period of the process of the application. Therefore, the execution of the process of the application will not be influenced.

[0030] In the embodiment, since the startup procedure of the process of the CUI application will not be interrupted by other execution threads, it is convenient to determine the time when the process of the application is to respond to the user input by the WaitForIdleInput interface, and thus acquire the startup period of the process of the application.

[0031] As shown in Figure 3, in another embodiment, in a case of a Graphical User Interface (GUI) application such as a resource manager, in a startup procedure of a process of the application, it requires to not only load codes and other information of the application, but also display user interactive windows (including a main window and child windows) in a graphical manner. Therefore, there are multiple execution threads in the startup procedure of the process of this type of application. The step 102 described above may be realized by the apparatus for determining the performance of the process of the application. The step 102 may include steps B1-B3.

[0032] B1: monitoring whether a user interactive main window corresponding to the application is opened.

[0033] B2: sending a message to the main window corresponding to the application in the case that the user interactive main window corresponding to the application is opened. The main window is a window on top of the process of the application, and normally a window with the largest area.

[0034] The apparatus for determining the performance of the process of the application may call a message monitoring (SetWindowsHookEx) interface and an event monitoring (SetWinEventHook) interface to wait for a notification of a message or event of the application. In a case where the apparatus for determining the performance of the process of the application calls the message monitoring interface, the message monitoring interface may selectively monitor displaying messages of a window corresponding the application, and may also selectively monitor other messages. When some messages are monitored, it is determined whether these messages are the displaying messages of the user interactive main window. If these messages are the displaying messages of the user interactive main window, the main window corresponding to the application has performed the display. In a case where the apparatus for determining the performance of the process of the application calls the event

monitoring interface, the event monitoring interface may monitor a displaying event of the main window corresponding to the application and selectively monitor a displaying message of the main window corresponding to the application. When the displaying events are monitored, it is determined whether these displaying events are the displaying events of the user interactive main window. If these displaying events are the displaying events of the user interactive main window, the main window corresponding to the application has performed the display.

[0035] When it is monitored that the main window corresponding to the application has performed the display, the apparatus for determining the performance of the process of the application may send a message to the main window. The message may be any message. Specifically, the apparatus for determining the performance of the process of the application may send the message by calling a SendMessage interface. If the main window responds to the message, it means that the process of the application may also respond to the user input. In this way, a response time of the main window is taken as a time when the process of the application is to respond to the user input.

[0036] B3: determining the time when the process of the application is to respond to the user input as the response time to the sent message described above from the main window corresponding to the application.

[0037] Furthermore, when the user interactive main window corresponding to the application does not respond to the message sent by the apparatus for determining the performance of the process of the application in step B2 within a preset second timeout, the apparatus for determining the performance of the process of the application will stop performing the step 102 described above, that is, stop determining the time when the process of the application is to respond to the user input. Hence, in the startup procedure of the process of the application, if the process of the application cannot respond to the user input due to some reasons, such as some execution threads in an endless loop, the apparatus for determining the performance of the process of the application will stop to acquire the startup period of the process of the application. Therefore, the execution of the process of the application will not be influenced.

[0038] In the embodiment, since the startup procedure of the process of the GUI application includes multiple execution threads, the user interactive main window corresponding to the

application may not respond to the user input within a time period after being displayed. In this case, the process of the application is not started completely. In the embodiment, it is required to determine a time when the main window is to respond to the user input after being displayed, as the process startup ending time of the application. In this way, the startup period of the process of the application is acquired accurately.

[0039] An embodiment of the present disclosure also provides a computer system, i.e., the apparatus for determining the performance of the process of the application described above. The startup performance of the process of the application may be determined by units of the computer system according to the embodiment of the method described above. Figure 4 shows a schematic structural diagram of the computer system, and the computer system includes a process startup unit 10, a startup time acquiring unit 11, a response time determining 12 and a performance determining unit 13.

[0040] The process startup unit 10 is adapted to start a process of an application.

[0041] The startup time acquiring unit 11 is adapted to, when the process of the application is started by the process startup unit 11, acquire a process startup time of the application. The startup time acquiring unit 11 may call an API for acquiring the startup time to acquire a time of the process of the application started by the process startup unit 10.

[0042] The response time determining 12 is adapted to determine a time when the process of the application is to respond to a user input as a process startup ending time of the application.

[0043] The performance determining unit 13 is adapted to acquire a startup period (i.e. a difference between the process startup ending time t2 and a process startup time t1) of the application according to the process startup time acquired by the startup time acquiring unit 11 and the process startup ending time to determine a startup performance of the process of the application.

[0044] In the computer system according to the embodiment of the present disclosure, the startup time acquiring unit 11 acquires the process startup time of the application when the process startup unit 10 starts the process of the application. The response time determining unit 12 determines the time when the process of the application can respond to the user input. The performance determining unit 13 takes the time when the process of the application can

respond to the user input as the process startup ending time of the application; and acquire the startup period of the application according the process startup time and the process startup ending time to determine the startup performance of the process of the application. Therefore, even if some processes of applications do not have user interactive interfaces, there is no influence on the calculation of the startup period of the process of the application. Thus, the startup period of the process of the application is acquired accurately, and the startup performance of the process of the application is determined accurately.

[0045] Referring to Figure 5, in an embodiment, in a case where the application is a CUI application, the computer system may further include a first stopping unit 14 in addition to the structure shown in Figure 4, and the response time determining unit 12 may be implemented by an input calling unit 120 and a first time determining unit 121.

[0046] The input calling unit 120 is adapted to call a WaitForIdleInput interface to determine whether the process of the application started by the process startup unit 10 can respond to the user input.

[0047] The first time determining unit 121 is adapted to take a time when the WaitForIdleInput interface called by the input calling unit 120 provides a return value as the time when the process of the application can respond to the user input.

[0048] The first stopping unit 14 is adapted to, when the WaitForIdleInput interface called by the input calling unit 120 does not provide the return value within a preset first timeout, notify the response time determining unit 12 to stop determining the time when the process of the application can respond to the user input.

[0049] In the computer system according to the embodiment, the input calling unit 120 of the response time determining unit 12 may call the WaitForIdleInput interface. The first time determining unit 121 determines the time when the WaitForIdleInput interface provides a return value. The performance determining unit 13 takes the time determined by the first time determining unit 121 as the process startup ending time of the application. If the WaitForIdleInput interface does not provide the return value within a preset first timeout, the first stopping unit 14 will notify the response time determining unit 12 to stop determining the time when the process of the application can respond to the user input.

[0050] Referring to Figure 6, in an embodiment, in a case where the application is a GUI

application, the computer system may include a second stopping unit 15 in addition to the structure shown in Figure 4, and the response time determining unit 12 may be implemented by a message sending unit 122, a second time determining unit 123 and a monitoring unit 124.

[0051] The monitoring unit 124 is adapted to monitor whether a user interactive main window corresponding to the application is opened. Specifically, the monitoring unit 124 may call a message monitoring interface and an event monitoring interface to wait for a notification of a message or an event of the application. When the message monitoring interface or the event monitoring interface has received the notification of a message or event of the application, the monitoring unit 124 determines whether the message or event is a displaying message or displaying event of the user interactive main window. If the message or event is a displaying message or displaying event of the user interactive main window, the monitoring unit 124 monitors that the user interactive main window corresponding to the application is opened.

[0052] The message sending unit 122 is adapted to send a message to the main window corresponding to the application started by the process startup unit 10 when the monitoring unit 124 determines that the user interactive main window corresponding to the application is opened.

[0053] The second time determining unit 123 is adapted to take a response time that the main window corresponding to the application responds to the message sent by the message sending unit 122 as the time when the process of the application can respond to the user input.

[0054] The second stopping unit 15 is adapted to, when the main window does not respond to the message sent by the message sending unit 122 within a preset second timeout, stop determining the time when the process of the application can respond to the user input.

[0055] In the computer system according to the embodiment, the message sending unit 122 of the response time determining unit 12 may send a message when it is monitored that the user interactive main window corresponding to the application is opened. The second time determining unit 123 determines the response time for responding to the message from the opened main window. The performance determining unit 13 takes the time determined by the second time determining unit 123 as the process startup ending time of the application. If the main window does not respond to the message within the preset first timeout, the second stopping unit 15 will notify the response time determining unit 12 to stop determining the

time when the process of the application can respond to the user input.

[0056] In the following, for example, the method for determining a performance of a process of an application according to the embodiments of the present disclosure is applied to a terminal. The terminal may be a smart phone, a panel computer, an ebook reader, a Moving Picture Experts Group Audio Layer III (MP3) player, a Moving Picture Experts Group Audio Layer IV (MP4) player, a laptop portable computer, a desktop computer and so on.

[0057] Reference is made to Figure 7 which shows a schematic structural diagram of a terminal according to an embodiment of the present disclosure.

[0058] The terminal may include: a Radio Frequency (RF) circuit 20, a memory 21 including one or more computer readable storage mediums, an input unit 22, a display unit 23, a sensor 24, an audio circuit 25, a wireless fidelity (WiFi) module 26, a processor 27 including one or more processing cores, a power supply 28 and so on. Those skilled in the art may understand that the terminal is not limited to a structure of the terminal shown in Figure 7, and may include more components or less components than the illustrated components, a combination of the components, or a different arrangement of the components.

[0059] The RF circuit 20 may be adapted to receive and send information, or receive or send a signal in a calling procedure. Specifically, the RF circuit 20 receives downlink information from a base station, and sends the information to one or more processors 27 to process. In addition, the RF circuit 20 sends uplink data to the base station. Normally, the RF circuit 20 includes but not limited to an antenna, at least one amplifier, a tuner, one or more oscillators, a Subscriber Identity Module (SIM) card, a transceiver, a coupler, a Low Noise Amplifier (LNA), a duplexer and so on. Furthermore, the RF circuit 20 may also communicate with a network or other devices via a wireless communication. The wireless communication may be operated in any communication standard or protocol, which includes but not limited to a Global System of Mobile communication (GSM), a General Packet Radio Service (GPRS), a Code Division Multiple Access (CDMA), a Wideband Code Division Multiple Access (WCDMA), a Long Term Evolution (LTE), an e-mail, a Short Messaging Service (SMS) and so on.

[0060] The memory 21 may be adapted to store software programs or modules, and the processor 27 is adapted to perform various function application and data processing by running the software programs or modules stored in the memory 21. The memory 21 may

mainly include a program memory area and a data memory area. Specifically, the program memory area may store an operating system, an application required by at least one function (such as a sound playing function, an image playing function) and so on. The data memory area may store data (such as audio data, a phone book) created according to terminal utilization, and so on. Furthermore, the memory 21 may be a high speed random access memory, and may also be a nonvolatile memory such as at least one magnetic disk memory device, a flash memory device or other volatile solid state memory device. Accordingly, the memory 21 may also include a memory controller for providing access of the processor 27 and the input unit 22 to the memory 21.

[0061] The input unit 22 may be used to receive digital information or character information that is input, and generate signal inputs by a keyboard, a mouse, a joystick or a trackball, which relate to user settings and function control. Specifically, in an embodiment, the input unit 22 may include a touch-sensitive surface 221 and other input devices 222. The touch-sensitive surface 221, also referred to as a touch display screen or a touch panel, may collect touch operations operated by the user thereon or in the vicinity thereof, such as operations made by the user using any suitable object or accessory (such as a finger, a touch pen) on the touch-sensitive surface 221 or in the vicinity of the touch-sensitive surface 221; and then may drive a corresponding connection device according to a program set in advance. Optionally, the touch-sensitive surface 221 may include a touch detection device and a touch controller. Specifically, the touch detection device detects the touch position of the user and a signal caused by the touch operation, and sends the signal to the touch controller. The touch controller receives the touch information from the touch detection device, converts the information into contact coordinates, then sends the coordinates of the touch point to the processor 27, and receives a command sent from the processor 27 to perform. Furthermore, the touch-sensitive surface 221 may be realized in multiple types such as in resistive type, in capacitive type, in infrared type and in surface acoustic wave type. In addition to the touch-sensitive surface 221, the input unit 22 may also include other input devices 222. Specifically, the other input devices 222 may include but not limited to one or more of a physical keyboard, a function key (such as a volume control button, and a switch button), a trackball, a mouse, a joystick and so on.

[0062] The display unit 23 may be used to display information input by the user, information provided to the user and various graphic user interfaces of the terminal. Those

graphic user interfaces may include a graphic, a text, an icon, a video and any other combination thereof. The display unit 23 may include a display panel 231. Optionally, the display panel 231 may be configured as for example a Liquid Crystal Display (LCD) or an Organic Light-Emitting Diode (OLED) display. Furthermore, the touch-sensitive surface 221 may cover the display panel 231. When the touch-sensitive surface 221 detects a touch operation thereon or in the vicinity thereof, the touch-sensitive surface 221 sends the detected touch operation to the processor 27 to determine a type of the touch event. The processor 27 then provides a corresponding visual output on the display panel 231 according to the type of the touch event. The touch-sensitive surface 221 and the display panel 231, which is shown as two separate components to realize the input function and the output function respectively in Figure 5, may be integrated together to realize the input function and the output function in some embodiments.

[0063] The terminal may further include at least one sensor 24 such as a light sensor, a motion sensor and other sensors. Specifically, the light sensor may include an ambient light sensor and a proximity sensor. The ambient light sensor may adjust a brightness of the display panel 231 according to ambient light. The proximity sensor may turn off the display panel 231 and/or a backlight when the terminal is closed to the user's ear. As one of the motion sensors, a gravity acceleration sensor may detect an acceleration value in each direction (generally, in three axial directions), and detect a value and a direction of the gravity in a stationary state. The gravity acceleration sensor may be applied to an application (such as orientation change, related games, magnetometer attitude calibration) for identifying the attitude of a cell phone, a function related to vibration identification (such as a pedometer, or a knock) and so on. Other sensors such as a gyroscope, a barometer, a hygrometer, a thermometer, an infrared sensor, which may be equipped to the terminal, will not be described here any more.

[0064] An audio interface between the user and the terminal may be provided by the audio circuit 25, a speaker 161 and a microphone 162. The audio circuit 25 may convert the received audio data into an electrical signal and transmit the electrical signal to the speaker 161, and the speaker 161 converts the electrical signal into a sound signal to output. On the other hand, the microphone 162 converts the collected sound signal into an electrical signal and sends the electrical signal to the audio circuit 25, and the audio circuit 25 converts the received electrical signal into audio data and outputs the audio data to the processor 27 to process. The processed audio data is then sent to for example another terminal via the RF

circuit 20 or output to the memory 21 to further process. The audio circuit 25 may also include an earphone jack to provide communication between a peripheral headphone and the terminal.

[0065] The WiFi is a short range wireless transmission technology. The terminal may assist the user to send and receive e-mails, browse a webpage and access a streaming media and so on via the WiFi module 26. The WiFi module 26 provides a wireless broadband internet access to the user. Although the WiFi module 26 is illustrated in Figure 7, it may be understood that the WiFi module 26 is not necessary in the terminal, and can be omitted without changing the scope of essence of the present disclosure.

[0066] The processor 27, as a control center of the terminal, is adapted to connect each part of the terminal using various interfaces and lines, and performs various functions of the terminal and data processing by running or executing the software program and/or the software module stored in the memory 21 and calling data stored in the memory 21, thus achieving the monitor of the whole terminal. Optionally, the processor 27 may include one or more processing cores. Preferably, an application processor and a modem processor may be integrated into the processor 27. In the process 27, the application processor mainly processes the operating system, user interfaces, applications and so on; and the modem processor mainly processes wireless communication. It may be understood that the modem processor described above may not be integrated into the processor 27.

[0067] The terminal may further include the power supply 28 (e.g. a battery) for supplying power to each component. Preferably, the power source may be logically connected to the processor 27 via a power supply management system, thus achieving functions such as charge management, discharge management, power consumption management by the power supply management system. The power supply 28 may further include any components, such as one or more DC power supplies or AC power supplies, a recharge system, a power supply fault detection system, a power supply converter or inverter, or a power supply state indicator.

[0068] Although not illustrated, the terminal may also include a camera, a Bluetooth module and so on, which are not described here any more. In this embodiment, the processor 27 in the terminal will perform one or more applications stored in the memory 21 according to the following instructions to realize various functions. The instructions include the following steps:

acquiring a process startup time of an application in the case that a process of the application is started.

determining a time when the process of the application is to respond to a user input as a process startup ending time of the application; and

acquiring a startup period of the application according the process startup time and the process startup ending time to determine a startup performance of the process of the application.

[0069] Furthermore, in a case where the application is a CUI application, in the step of determining the time when the process of the application is to respond to the user input, the processor 27 may call a WaitForIdleInput interface to determine whether the process of the application can respond to the user input, and takes a time when the WaitForIdleInput interface provides a return value as the time when the process of the application is to respond to the user input. When the WaitForIdleInput interface does not provide the return value within the preset first timeout, the processor 27 stops the step of determining the time when the process of the application is to respond to the user input.

[0070] On the other hand, in a case where the application is a GUI application, in the step of determining the time when the process of the application is to respond to the user input, the processor 27 will monitor whether the user interactive main window corresponding to the application is opened; in the case that the user interactive main window corresponding to the application is opened, send a message to the main window corresponding to the application; and take the response time of the main window corresponding to the application as the time when the process of the application is to respond to the user input. When the main window does not respond to the message within a preset second timeout, the processor 27 stops the step of determining the time when the process of the application is to respond to the user input.

[0071] Specifically, in the step of monitoring whether the user interactive main window corresponding to the application is opened, the processor 27 may call a message monitoring interface or an event monitoring interface to wait for a notification of a message or event of the application. When the message monitoring interface or the event monitoring interface has received the notification of a message or event of the application, the processor 27 determines whether the message or the event is a displaying message or a displaying event of the user

interactive main window. If the message or the event is the displaying message or the displaying event of the user interactive main window, the user interactive main window corresponding to the application is opened.

[0072] Those skilled in the art should understand that all or a part of steps in various methods in the embodiments described above can be implemented by instructing related hardware using a program. The program may be stored in a computer readable storage medium. The storage medium may include a Read-Only Memory (ROM), a Random Access Memory (RAM), a magnetic disk, an optical disk or the like.

[0073] Hereinbefore, the method for determining the performance of the process of the application and the computer system are introduced in detail above according to the embodiments of the present disclosure. The principle and the embodiments of the present disclosure have been interpreted by specific examples used herein, and the above description of the embodiments is just used to assist with understanding the method and the core concept of the present disclosure. Meanwhile, those skilled in the art may make some variations in the embodiments and the application scope according to the concept of the present disclosure. In summary, the content of this description should not be interpreted as limiting the disclosure.

CLAIMS

1. A method for determining a performance of a process of an application, comprising:

acquiring a process startup time of the application in the case that the process of the application is started;

determining a time when the process of the application is to respond to a user input as a process startup ending time of the application; and

acquiring a startup period of the application according to the process startup time and the process startup ending time to determine a startup performance of the process of the application.

2. The method according to claim 1, wherein in a case where the application is a CUI application, the determining a time when the process of the application is to respond to a user input comprises:

calling a WaitForIdleInput interface to determine whether the process of the application is to respond to the user input; and

taking a time when the WaitForIdleInput interface provides a return value as the time when the process of the application is to respond to the user input.

3. The method according to claim 2, wherein in the case that the WaitForIdleInput interface does not provide the return value within a preset first timeout, the determining a time when the process of the application is to respond to a user input is stopped.

4. The method according to claim 1, wherein in a case where the application is a GUI application, the determining a time when the process of the application is to respond to a user input comprises:

monitoring whether a user interactive main window corresponding to the application is opened;

sending a message to the user interactive main window corresponding to the application in the case that the user interactive main window corresponding to the application is opened; and

taking a response time of the user interactive main window corresponding to the application as the time when the process of the application is to respond to the user input.

5. The method according to claim 4, wherein the monitoring whether a user interactive main window corresponding to the application is opened comprises:

calling a message monitoring interface to wait for a notification of a message of the application, or calling an event monitoring interface to wait for a notification of a event of the application; and

in the case that the message monitoring interface receives the notification of the message of the application or in the case that the event monitoring interface receives the notification of the event of the application, determining that the message is a displaying message of the user interactive main window or the event is the a displaying event of the user interactive main window; and monitoring that the user interactive main window corresponding to the application is opened.

6. The method according to claim 4 or 5, wherein in the case that the main window does not respond to the message within a preset second timeout, the determining a time when the process of the application is to respond to a user input is stopped.

7. A computer system, comprising:

a process startup unit adapted to start a process of an application;

a startup time acquiring unit adapted to acquire a process startup time of the application in the case that the process of the application is started by the process startup unit;

a response time determining unit adapted to determine a time when the process of the application is to respond to a user input as a process startup ending time of the application; and

a performance determining unit adapted to acquire a startup period of the application according to the process startup time and the process startup ending time to determine a startup performance of the process of the application.

8. The computer system according to claim 7, wherein in a case where the application is a CUI application, the response time determining unit comprises:

an input calling unit adapted to call a WaitForIdleInput interface to determine whether the process of the application is to respond to the user input; and

a first time determining unit adapted to take a time when the WaitForIdleInput interface provides a return value as the time when the process of the application is to respond to the user input.

9. The computer system according to claim 8, further comprising:

a first stopping unit adapted to, in the case that the WaitForIdleInput interface does not provide the return value within a preset first timeout, notify the response time determining unit to stop determining the time when the process of the application is to respond to the user input.

10. The computer system according to claim 7, wherein in a case where the application is a GUI application, the response time determining unit comprises:

a monitoring unit adapted to monitor whether a user interactive main window corresponding to the application is opened;

a message sending unit adapted to send a message to the user interactive main window corresponding to the application, in the case that the user interactive main window corresponding to the application is opened; and

a second time determining unit adapted to take a response time of the user interactive main window corresponding to the application as the time when the process of the application is to respond to the user input.

11. The computer system according to claim 10, wherein

the monitoring unit is adapted to call a message monitoring interface to wait for a notification of a message of the application, or call an event monitoring interface to wait for a notification of a event of the application; and in the case that the message monitoring interface receives the notification of the message of the application or in the case that the event monitoring interface receives the notification of the event of the application, determine that the message is a displaying message of the user interactive main window or the event is the a displaying event of the user interactive main window; and monitor that the user interactive main window corresponding to the application is opened.

12. The computer system according to claim 10 or 11, further comprising:

a second stopping unit adapted to, in the case that the main window does not respond to the message within a preset second timeout, stop determining the time when the process of the application is to respond to the user input.

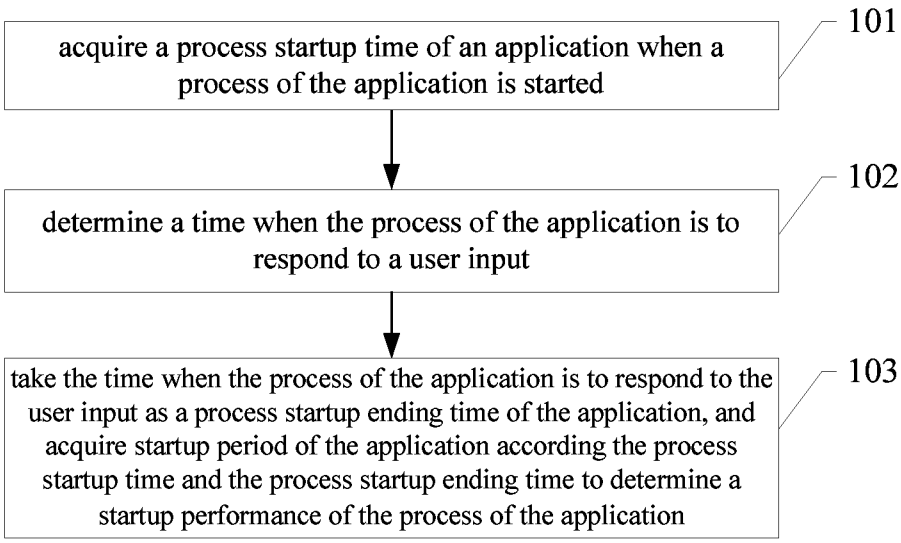
acquire a process startup time of an application when a process of the application is started
101
determine a time when the process of the application is to respond to a user input
102
take the time when the process of the application is to respond to the user input as a process startup ending time of the application, and acquire startup period of the application according the process startup time and the process startup ending time to determine a startup performance of the process of the application
103

Fig. 1

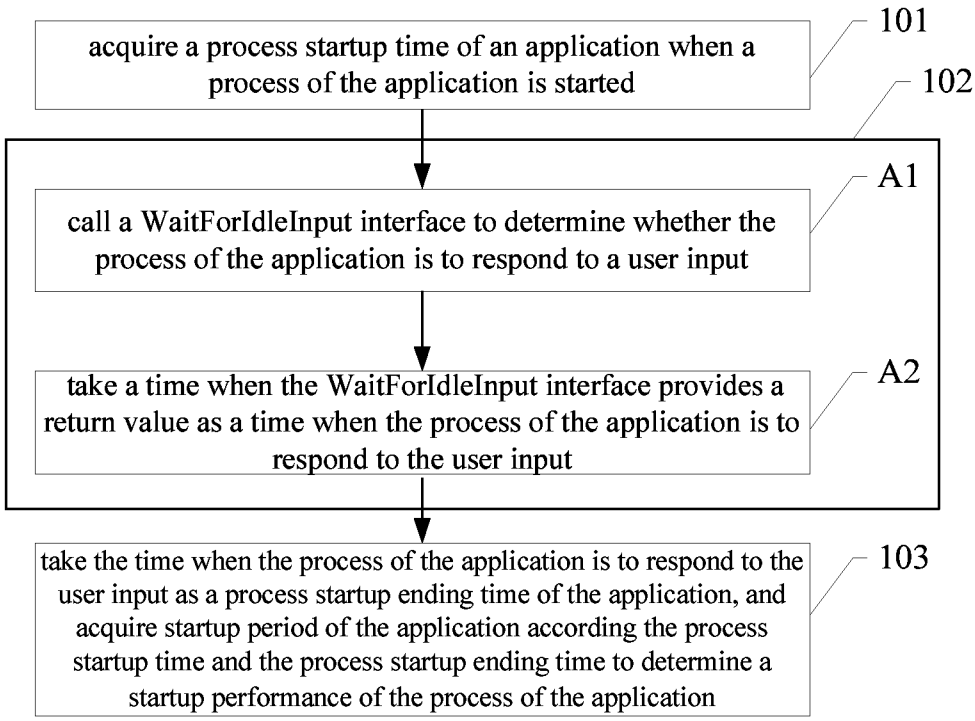
acquire a process startup time of an application when a process of the application is started
101
102
call a WaitForIdleInput interface to determine whether the process of the application is to respond to a user input
A1
take a time when the WaitForIdleInput interface provides a return value as a time when the process of the application is to respond to the user input
A2
take the time when the process of the application is to respond to the user input as a process startup ending time of the application, and acquire startup period of the application according the process startup time and the process startup ending time to determine a startup performance of the process of the application
103

Fig. 2

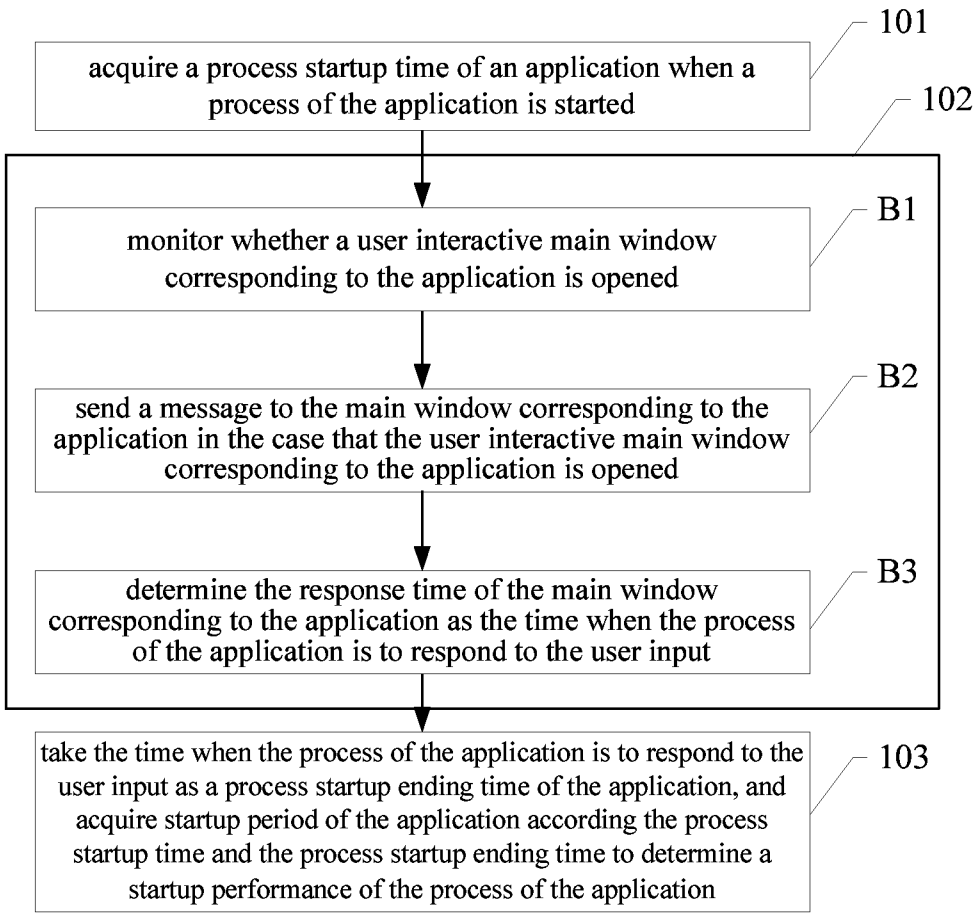
101
acquire a process startup time of an application when a process of the application is started
102
B1
monitor whether a user interactive main window corresponding to the application is opened
B2
send a message to the main window corresponding to the application in the case that the user interactive main window corresponding to the application is opened
B3
determine the response time of the main window corresponding to the application as the time when the process of the application is to respond to the user input
103
take the time when the process of the application is to respond to the user input as a process startup ending time of the application, and acquire startup period of the application according the process startup time and the process startup ending time to determine a startup performance of the process of the application

Fig. 3

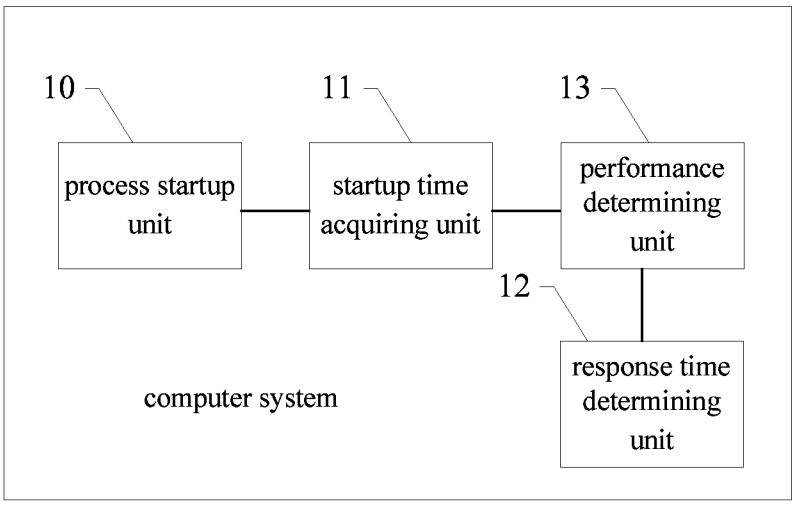
10
11
13
process startup unit
startup time acquiring unit
performance determining unit
12
response time determining unit
computer system

Fig. 4

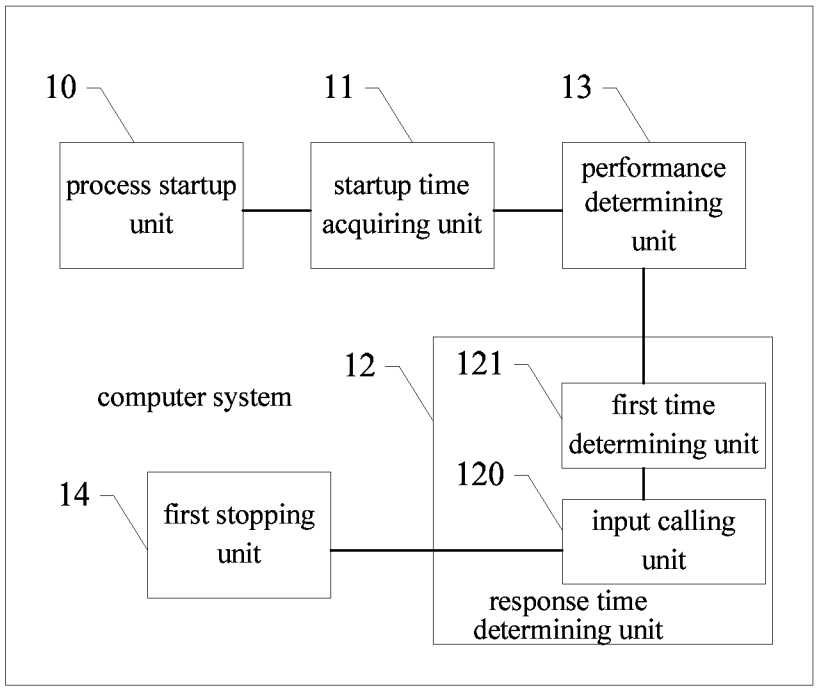
10
11
13
process startup unit
startup time acquiring unit
performance determining unit
12
121
computer system
first time determining unit
120
14
first stopping unit
input calling unit
response time determining unit

Fig. 5

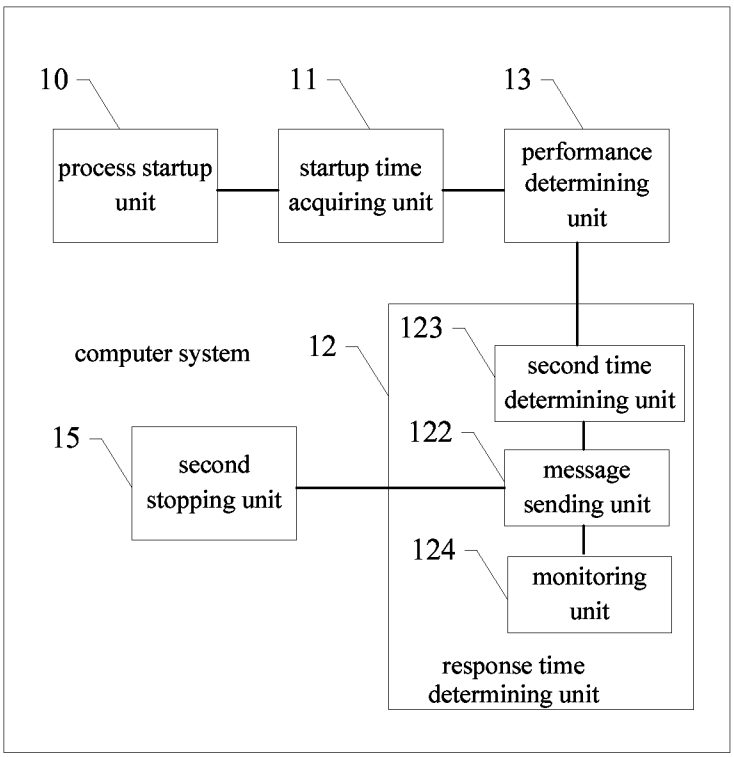
10
process startup unit
11
startup time acquiring unit
13
performance determining unit
computer system
12
123
second time determining unit
122
message sending unit
124
monitoring unit
response time determining unit
15
second stopping unit

Fig. 6

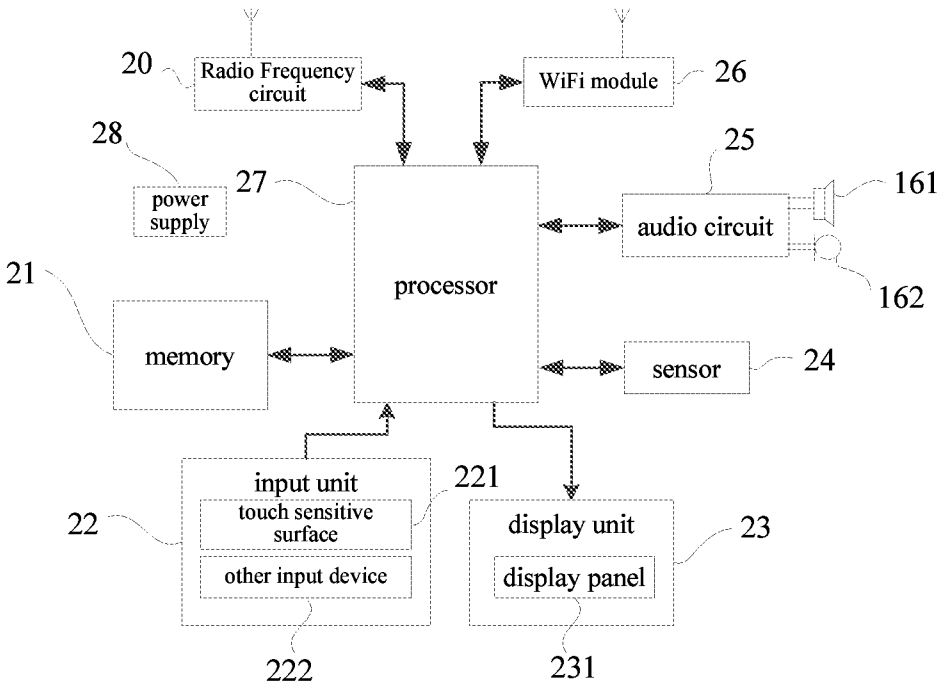
20
Radio Frequency circuit
WiFi module
26
28
power supply
27
25
audio circuit
161
processor
21
memory
sensor
24
162
input unit
221
touch sensitive surface
other input device
22
display unit
display panel
23
222
231

Fig. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2014/070763

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/445(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 9/-

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)

CNKI,CNABS,CNTXT,VEN:thread+,process+,application?,software,start+,end+,intializ+,load+,finish+,over,time?,respond +,user input,message?,event?,window

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 101000575 A ((YINGYEDA CO., LTD.)) 18 July 2007 (2007-07-18) description, page 4 paragraph 4	1-3, 7-9
Y	CN 101000575 A ((YINGYEDA CO., LTD.)) 18 July 2007 (2007-07-18) description, page 4 paragraph 4	4-6, 10-12
Y	CN 103034915 A ((SHENZHEN KINGDEE YOUSHANG ELECTRONIC COMMERCE SERVICE CO., LTD.)) 10 April 2013 (2013-04-10) description, paragraphs 0079-0093	4-6, 10-12
PX	CN 103197944 A ((TENCENT TECH SHENZHEN CO., LTD.)) 10 July 2013 (2013-07-10) the whole document	1-12

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

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
10 April 2014	**14 May 2014**

Name and mailing address of the ISA/	Authorized officer
STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA(ISA/CN) **6,Xitucheng Rd., Jimen Bridge, Haidian District, Beijing, China** **100088 China**	**MA,Yafan**
Facsimile No. **(86-10)62019451**	Telephone No. **(86-10)62089420**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2014/070763

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 101000575 A	18 July 2007	None	
CN 103034915 A	10 April 2013	None	
CN 103197944 A	10 July 2013	None	