



(19) **United States**

(12) **Patent Application Publication**
Han et al.

(10) **Pub. No.: US 2023/0385131 A1**

(43) **Pub. Date: Nov. 30, 2023**

(54) **METHOD FOR ACCELERATING APPLICATION STARTUP, ELECTRONIC DEVICE, AND COMPUTER STORAGE MEDIUM**

Publication Classification

(51) **Int. Cl.**
G06F 9/54 (2006.01)
G06F 9/445 (2006.01)
(52) **U.S. Cl.**
CPC *G06F 9/542* (2013.01); *G06F 9/546* (2013.01); *G06F 9/44521* (2013.01)

(71) Applicant: **Huawei Technologies Co., Ltd.**,
Shenzhen (CN)

(72) Inventors: **Feng Han**, Shenzhen (CN); **Wenyong Sun**, Shenzhen (CN)

(57) **ABSTRACT**

A system service detects a first trigger event of a target application, loads an application program of the target application in response to the first trigger event, and transmits a communication address to the application program. The application program performs preloading, and waits to receive a message based on the communication address. The system service detects a second trigger event of the target application. The system service generates a startup instruction in response to the second trigger event of the target application, and sends the startup instruction to the application program. The application program receives the startup instruction by using the communication address, and starts the target application in response to the startup instruction.

(21) Appl. No.: **18/043,194**

(22) PCT Filed: **Aug. 4, 2021**

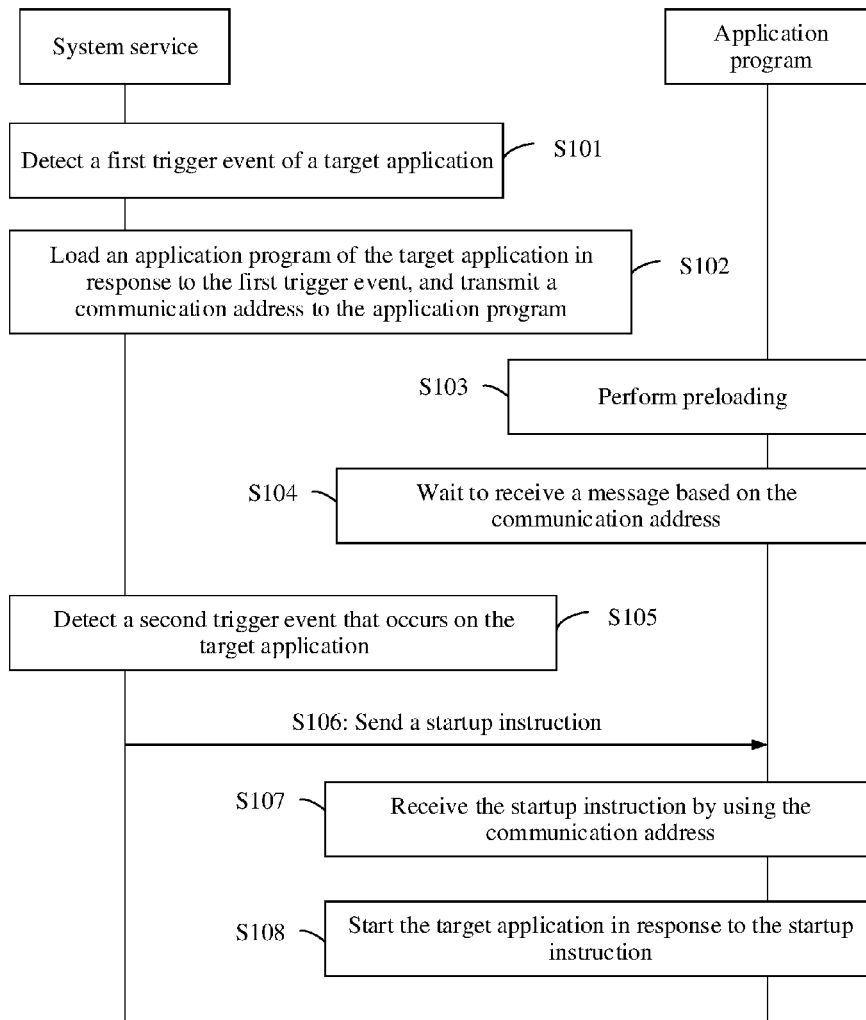
(86) PCT No.: **PCT/CN2021/110463**

§ 371 (c)(1),

(2) Date: **Feb. 27, 2023**

(30) **Foreign Application Priority Data**

Aug. 31, 2020 (CN) 202010901120.4



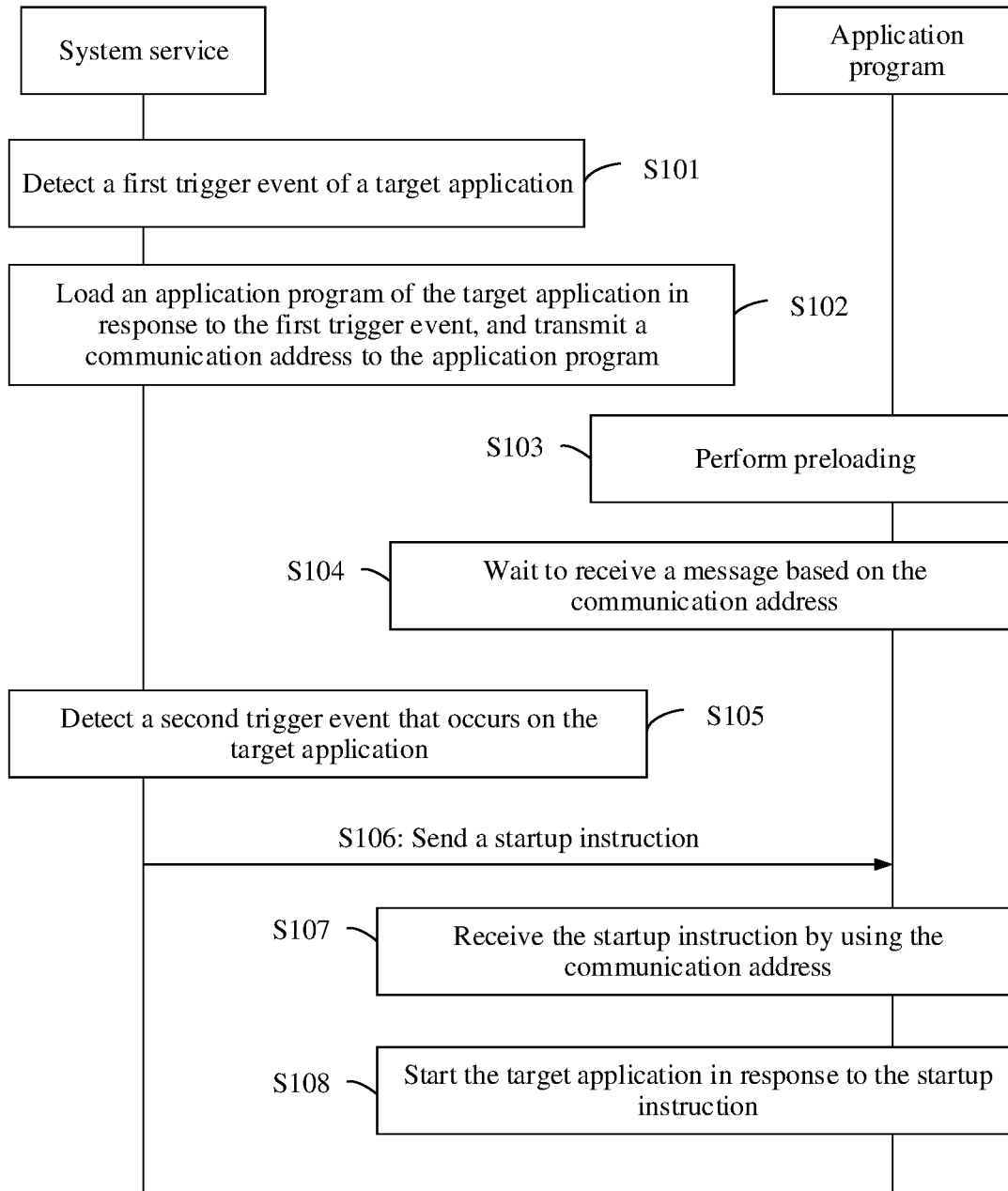


FIG. 1

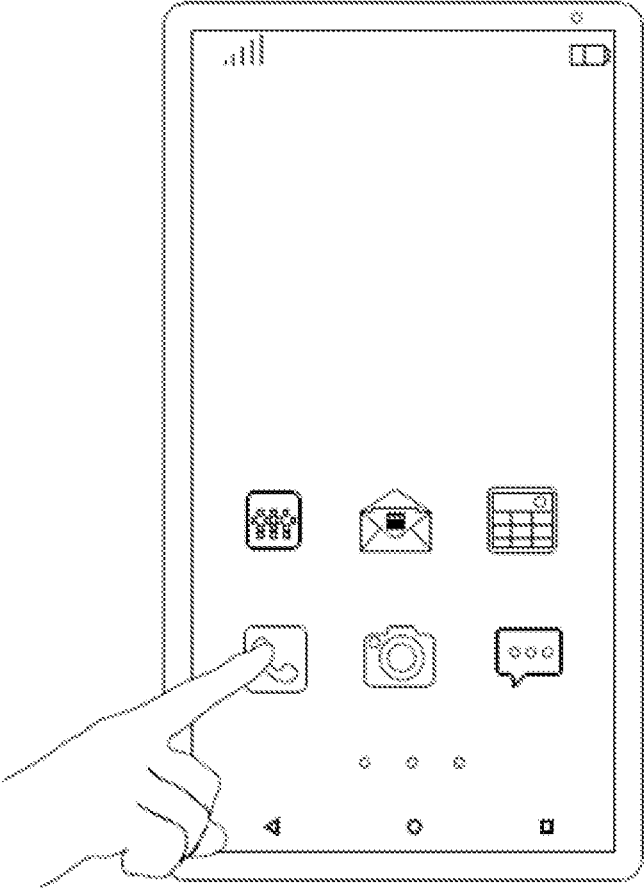


FIG. 2A

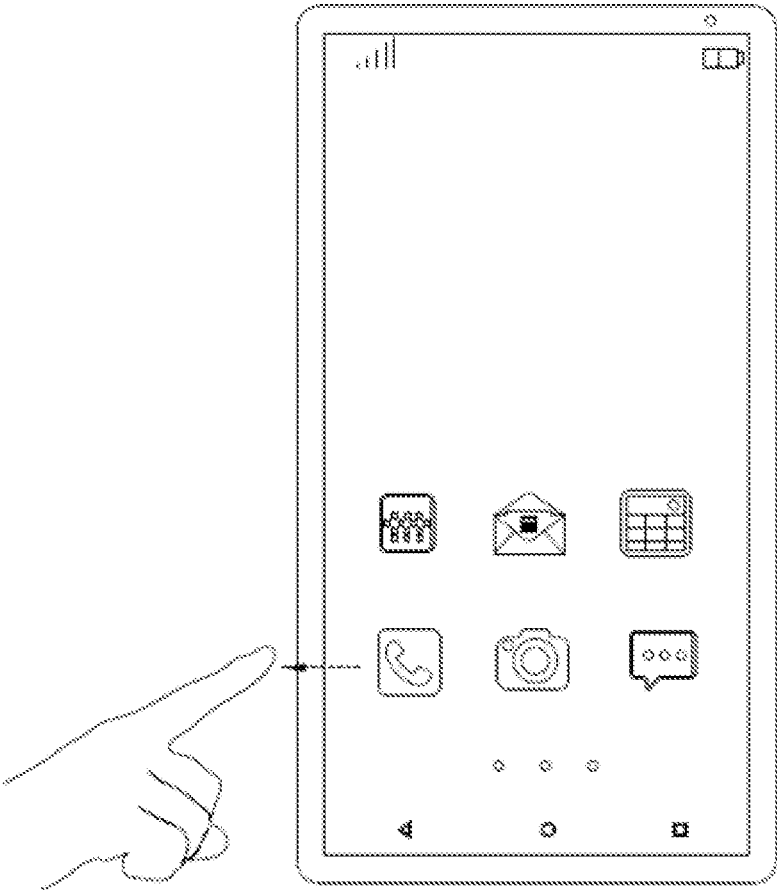


FIG. 2B

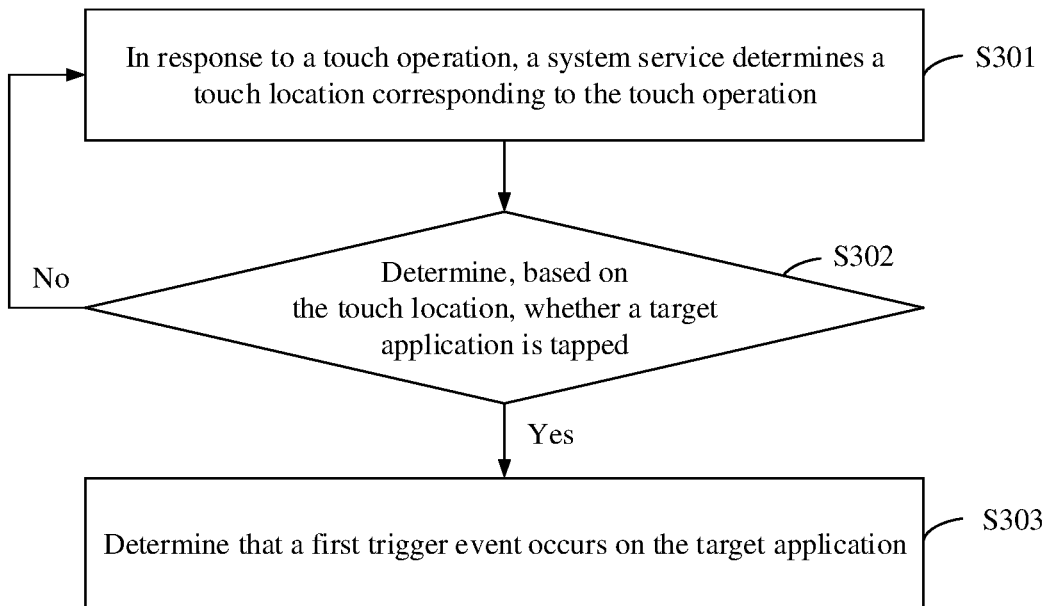


FIG. 3

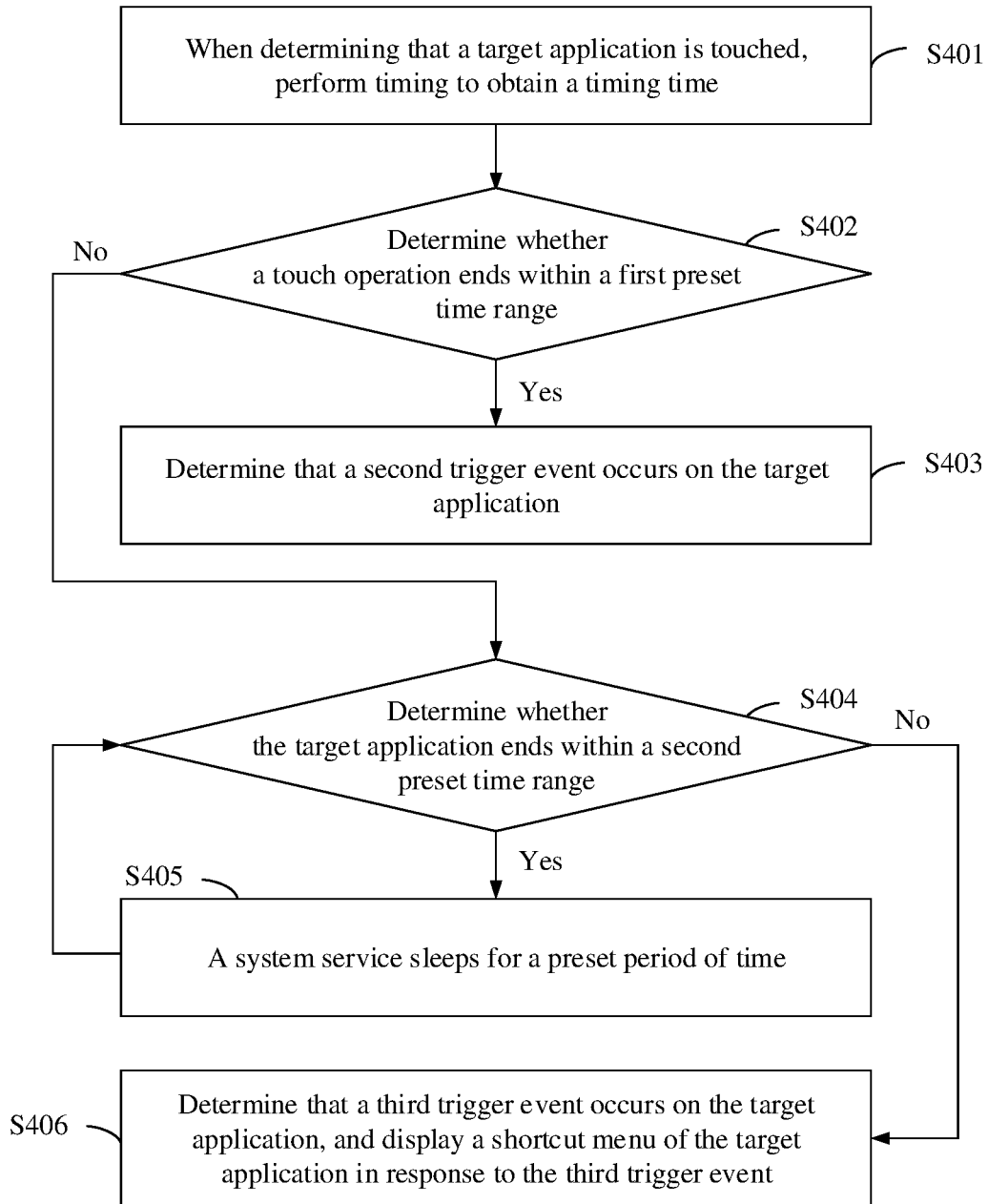


FIG. 4

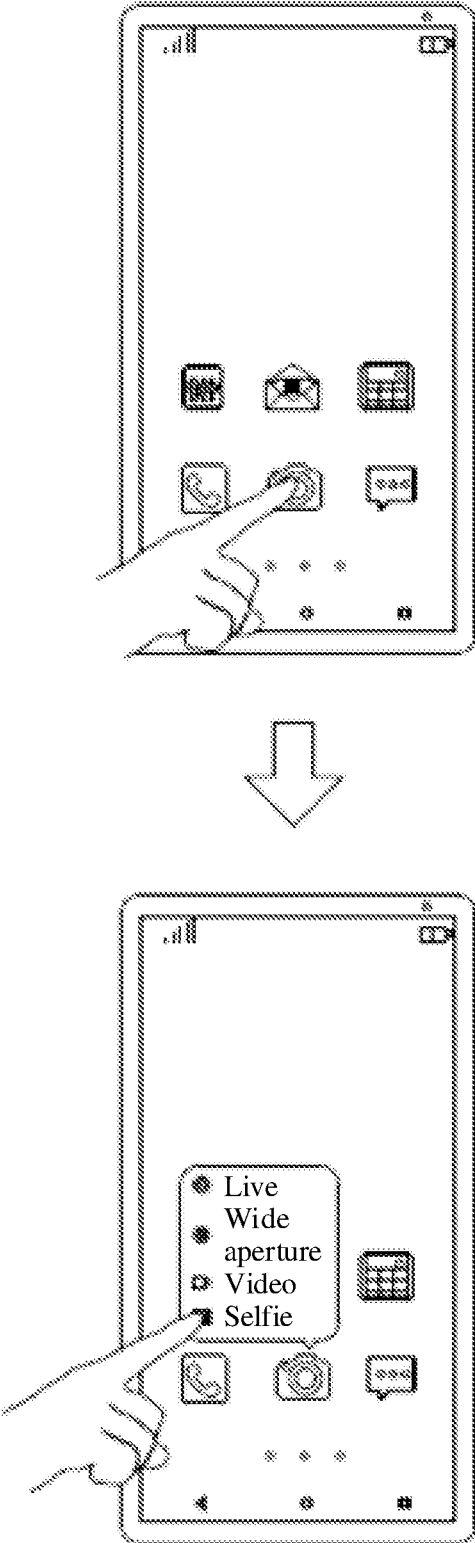


FIG. 5

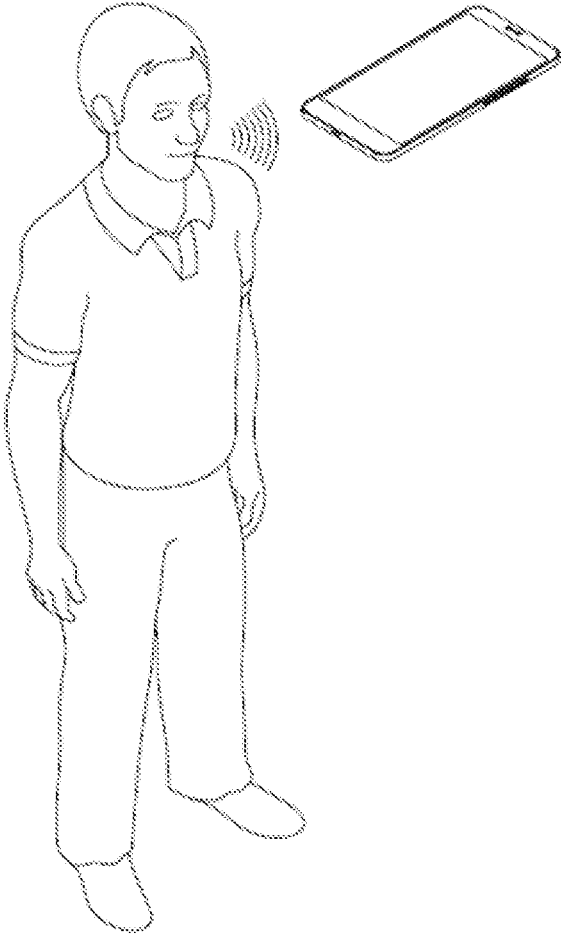


FIG. 6

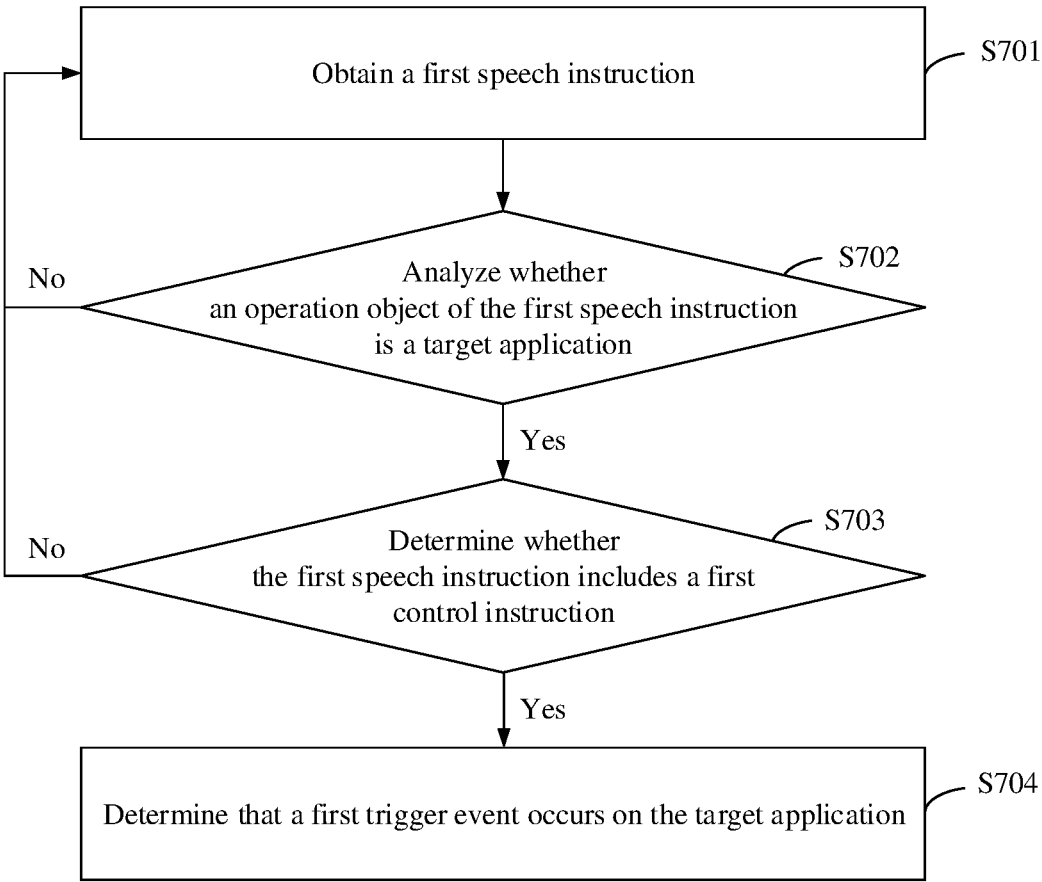


FIG. 7

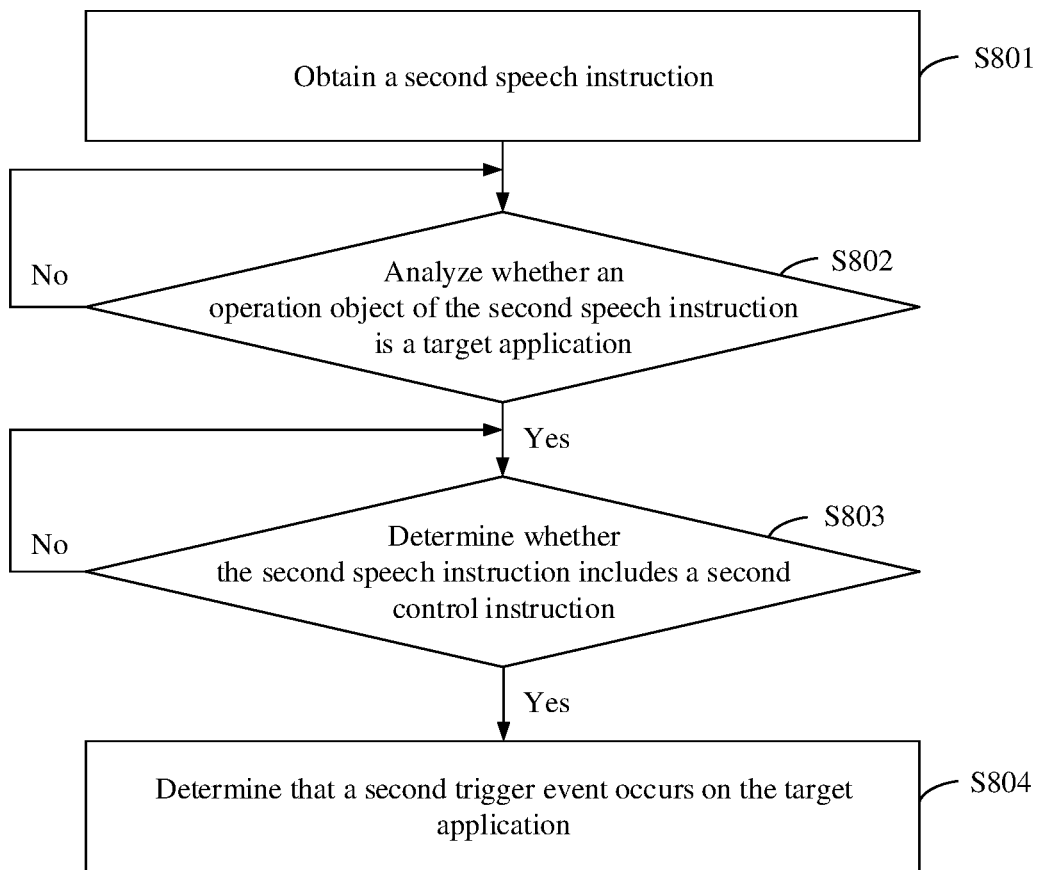


FIG. 8

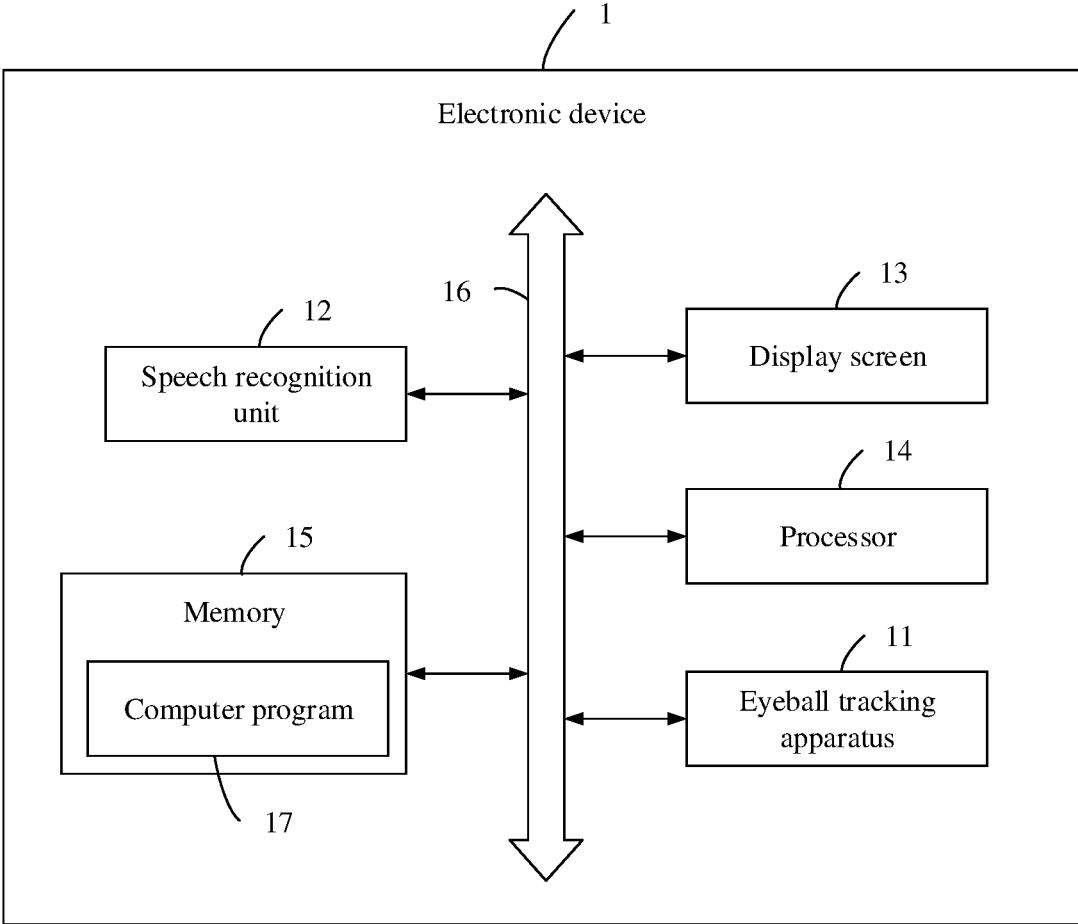


FIG. 9

**METHOD FOR ACCELERATING
APPLICATION STARTUP, ELECTRONIC
DEVICE, AND COMPUTER STORAGE
MEDIUM**

CROSS-REFERENCE TO RELATED
APPLICATIONS

[0001] This application is a National Stage of International Application No. PCT/CN2021/110463, filed on Aug. 4, 2021, which claims priority to Chinese Patent Application No. 202010901120.4, filed on Aug. 31, 2020, both of which are hereby incorporated by reference in their entireties.

TECHNICAL FIELD

[0002] This application relates to the field of mobile terminal technologies, and in particular, to a method for accelerating application startup, an electronic device, and a computer storage medium.

BACKGROUND

[0003] With widespread application of mobile terminals such as smartphones, smartphones can support an increasing quantity of applications, have increasingly powerful functions, develop toward diversification and personalization, and have become indispensable electronic products in users' lives. However, a startup speed of an application on a smartphone has become an important factor that affects consumer experience and media evaluation. Currently, in the conventional technology, when an application is used, the application is started in advance and placed in the background. To be specific, after a smartphone is powered on, an application is started and is set to run in the background, and after a user touches the application, a background program only needs to be set to a foreground application. However, in the foregoing solution, an application program is executed when the application is started, and the application program receives all messages in the background. As a result, data processing by the application program is uncontrollable, and memory usage is excessively large when the application program loads data.

SUMMARY

[0004] In view of the foregoing content, a method for accelerating application startup, an electronic device, and a computer storage medium need to be provided, to make a preloading behavior of an application program controllable and avoid excessively large memory usage when the application program loads data.

[0005] According to a first aspect, this application provides a method for accelerating application startup, specifically including:

[0006] a system service detects a first trigger event of a target application;

[0007] the system service loads an application program of the target application in response to the first trigger event, and transmits a communication address to the application program; the application program performs preloading;

[0008] the application program waits to receive a message based on the communication address;

[0009] the system service detects a second trigger event of the target application;

[0010] the system service generates a startup instruction in response to the second trigger event of the target application, and sends the startup instruction to the application program;

[0011] the application program receives the startup instruction by using the communication address; and

[0012] the application program starts the target application in response to the startup instruction.

[0013] In this embodiment of this application, when the application program detects, by using the communication address, the startup instruction sent by the system service, the application program starts the target application in response to the startup instruction, so that socket blocking is implemented when the application program starts the target application. Therefore, in a preloading process of the application program of the target application before the application program of the target application receives the startup instruction, no excessive CPU resources, input/output resources, or memory space needs to be occupied, thereby ensuring controllable preloading behavior of the target application and increasing an application startup speed.

[0014] In a possible design, the method further includes: that the application program waits to receive a message based on the communication address includes:

[0015] the application program creates a server or a client based on the communication address; and

[0016] the server or the client waits to receive a message based on the communication address.

[0017] According to the foregoing technical solution, the application program may receive, by using the server or the client, a message that waits to be received based on the communication address.

[0018] In a possible design, the communication address is a socket address, the server is a socket server, and the client is a socket server.

[0019] According to the foregoing technical solution, the system service and the application program implement a socket communication mechanism.

[0020] In a possible design, the method further includes:

[0021] before the server or the client receives the startup instruction, the application program suspends execution of application code of the target application, and waits to receive the startup instruction;

[0022] the application program skips creating a child thread service; and

[0023] the application program skips initializing binder communication.

[0024] According to the foregoing technical solution, in the preloading process of the application program of the target application before the application program of the target application receives the startup instruction, usage of CPU resources and input/output resources is further reduced, thereby ensuring controllable preloading behavior of the target application and increasing an application startup speed.

[0025] In a possible design, that the system service sends the startup instruction to the application program specifically includes: The system service sends the startup instruction to the application program by using a socket message.

[0026] According to the foregoing technical solution, the application program can receive an instruction sent by the system server.

[0027] In a possible design, that the application program receives the startup instruction by using the server or the client specifically includes:

[0028] the server or the client in the application program listens to and receives, by using an accept() function and the communication address, the startup instruction sent by the system service.

[0029] According to the foregoing technical solution, a socket communication mechanism between an application service process and the system service is implemented.

[0030] In a possible design, that the application program starts the target application in response to the startup instruction specifically includes:

[0031] the application program invokes and executes the application code of the target application by using an oncreat() function.

[0032] According to the foregoing technical solution, the application code of the target application may be quickly executed.

[0033] In a possible design, that a system service detects a first trigger event of a target application includes: When detecting a touch operation on the target application, the system service determines that the first trigger event occurs on the target application.

[0034] According to the foregoing technical solution, when it is detected that a user performs a touch operation on the target application, the first trigger event may be triggered, to preload the target application, thereby facilitating an operation performed by the user and increasing a startup speed of the target application.

[0035] In a possible design, that a system service detects a first trigger event of a target application specifically includes:

[0036] in response to the touch operation, determining, by the system service, a touch location corresponding to the touch operation;

[0037] determining, based on the touch location, whether the target application is touched; and

[0038] when the target application is touched, determining that the first trigger event occurs on the target application.

[0039] According to the foregoing technical solution, the user can trigger the first trigger event only by performing a touch operation on a corresponding location of the target application on a display screen of an electronic device, to preload the target application, thereby facilitating an operation performed by the user and increasing a startup speed of the target application.

[0040] In a possible design, that the system service detects a second trigger event of the target application includes:

[0041] when detecting the first trigger event of the target application and then releasing the trigger operation, the system service determines that the second trigger event occurs on the target application.

[0042] According to the foregoing technical solution, the user only needs to perform the trigger operation on the target application and then release the trigger operation, so that the target application triggers the second trigger event, to start the target application, thereby facilitating an operation performed by the user.

[0043] In a possible design, that the system service detects a second trigger event of the target application specifically includes:

[0044] when determining that the target application is touched, performing, by the system service, timing to obtain a timing time;

[0045] determining whether the touch operation on the target application ends within a first preset time range; and

[0046] if the touch operation on the target application ends within the first preset time range, determining that the second trigger event occurs on the target application.

[0047] According to the foregoing technical solution, the user can trigger the second trigger event only by stopping the touch operation within the first preset time range after performing the touch operation on the target application, to start the target application, thereby facilitating an operation performed by the user.

[0048] In a possible design, the method further includes:

[0049] if the touch operation on the target application does not end within the first preset time range, the system service determines whether the target application ends within a second preset time range; and

[0050] if the touch operation ends within the second preset time range, the system service sleeps for a preset period of time; or

[0051] if the touch operation does not end within the second preset time range, the system service determines that a third trigger event occurs on the target application, and displays a shortcut menu of the target application in response to the third trigger event.

[0052] According to the foregoing technical solution, when a trigger operation performed by the user on the target application exceeds the second preset time range, a shortcut menu function of the target application is displayed, so that the user can perform an operation on a shortcut function of the target application.

[0053] In a possible design, that a system service detects a first trigger event of a target application specifically includes:

[0054] obtaining, by the system service, a first speech instruction;

[0055] analyzing whether an operation object of the first speech instruction is the target application;

[0056] if the operation object of the first speech instruction is the target application, determining whether the first speech instruction includes a first control instruction; and

[0057] if it is determined that the first speech instruction includes the first control instruction, determining that the first trigger event occurs on the target application.

[0058] According to the foregoing technical solution, the user can trigger the first trigger event only by using a speech instruction, to preload the target application, thereby facilitating an operation performed by the user and increasing a startup speed of the target application.

[0059] In a possible design, that the system service detects a second trigger event of the target application specifically includes:

[0060] obtaining, by the system service, a second speech instruction;

[0061] analyzing whether an operation object of the second speech instruction is the target application;

[0062] if the operation object of the second speech instruction is the target application, determining

- whether the second speech instruction includes a second control instruction; and
- [0063] if it is determined that the second speech instruction includes the second control instruction, determining that the second trigger event occurs on the target application.
- [0064] According to the foregoing technical solution, the user can trigger the second trigger event only by using a speech instruction, to start the target application, thereby facilitating an operation performed by the user and improving user experience.
- [0065] In a possible design, that a system service detects a first trigger event of a target application specifically includes:
- [0066] the system service obtains eyeball tracking information of a user; and
- [0067] the system service determines, based on the eyeball tracking information, that the first trigger event occurs on a target application whose gaze duration exceeds first preset duration.
- [0068] According to the foregoing technical solution, the user can trigger the first trigger event of the target application only by gazing at the target application with eyeballs, to preload the target application, thereby simplifying an operation performed by the user and increasing a startup speed of the target application.
- [0069] In a possible design, that the system service detects a second trigger event of the target application specifically includes:
- [0070] determining, based on the eyeball tracking information, whether gaze duration of the target application exceeds second preset duration; and
- [0071] if the gaze duration of the target application exceeds the second preset duration, determining that the second trigger event occurs on the target application.
- [0072] According to the foregoing technical solution, the user can trigger the second trigger event of the target application only by gazing at the target application with eyeballs for longer than the second preset duration, to preload the target application, thereby simplifying an operation performed by the user and increasing a startup speed of the target application.
- [0073] In a possible design, that a system service detects a first trigger event of a target application specifically includes:
- [0074] the system service obtains a location of an electronic device;
- [0075] the system service determines whether the location of the electronic device is within a preset area range; and
- [0076] if it is determined that the location of the electronic device is within the preset area range, the system service determines that the first trigger event occurs on the target application on the electronic device.
- [0077] According to the foregoing technical solution, when a user carrying the electronic device appears within the preset area range, the first trigger event of the target application can be triggered, to preload the target application.
- [0078] In a possible design, that a system service detects a first trigger event of a target application specifically includes:
- [0079] the system service determines whether a current time is within a preset time range; and
- [0080] if it is determined that the current time is within the preset time range, the system service determines that the first trigger event occurs on the target application.
- [0081] According to the foregoing technical solution, when the current time is within the preset time range, the first trigger event of the target application can be triggered, to preload the target application.
- [0082] In a possible design, that the system service detects a second trigger event of the target application specifically includes:
- [0083] when determining that the first trigger event occurs on the target application, the system service performs timing to obtain a timing time;
- [0084] the system service determines whether the timing time is greater than a second preset time threshold; and
- [0085] if the system service determines that the timing time is greater than the second preset time threshold, the system service determines that the second trigger event occurs on the target application.
- [0086] According to the foregoing technical solution, when the timing time after the first trigger event occurs reaches the second preset time threshold, the second trigger event of the target application is triggered, to start the target application.
- [0087] In a possible design, the target application is an application of a preset type. According to the foregoing technical solution, a target application of a specific type may trigger the first trigger event, to preload the target application.
- [0088] In a possible design, that a system service detects a first trigger event of a target application specifically includes:
- [0089] the system service obtains first historical data of using the target application by a user, where the first historical data includes at least time information, frequency information, and location information of using the target application;
- [0090] the system service analyzes the first historical data by using a machine learning method, and determines a usage habit condition of the user for the target application;
- [0091] the system service determines whether the target application meets the usage habit condition; and
- [0092] if it is determined that the target application meets the usage habit condition, the system service determines that the first trigger event occurs on the target application.
- [0093] According to the foregoing technical solution, when the target application meets the usage habit condition, the first trigger event of the target application can be triggered, to preload the target application.
- [0094] In a possible design, that the system service generates a preload instruction in response to the first trigger event, to preload the target application specifically includes:
- [0095] when the system service detects that the target application includes a plurality of applications, obtaining second historical data of the plurality of target applications;

[0096] determining a high-frequency usage time period or usage frequency of the plurality of target applications based on the second historical data of the plurality of target applications;

[0097] sorting the plurality of target applications based on a high-frequency usage time period or usage frequency of each target application; and

[0098] sequentially preloading the plurality of application programs based on the sorting.

[0099] According to the foregoing technical solution, when the first trigger event occurs on the plurality of target applications, the system service may preload a target application program with a higher level based on priority levels, thereby facilitating usage by the user.

[0100] According to a second aspect, an embodiment of this application provides an electronic device, including an apparatus for performing the steps performed in the method according to any one of the first aspect or the possible designs of the first aspect.

[0101] According to a third aspect, an embodiment of this application provides a computer storage medium. The computer storage medium stores program instructions. When the program instructions are run on an electronic device, the electronic device is enabled to perform the method for accelerating application startup according to any one of the first aspect or the possible designs of the first aspect in embodiments of this application.

[0102] In addition, for technical effects of the second aspect and the third aspect, refer to the descriptions of the methods in the designs of the foregoing method part. Details are not described herein again.

BRIEF DESCRIPTION OF THE DRAWINGS

[0103] FIG. 1 is a flowchart of a method for accelerating application startup according to an embodiment of this application;

[0104] FIG. 2A is a schematic diagram of a first trigger event occurring on a target application according to an embodiment of this application;

[0105] FIG. 2B is a schematic diagram of a second trigger event occurring on a target application according to an embodiment of this application;

[0106] FIG. 3 is a flowchart of a method for detecting a first trigger event of a target application by a system service according to an embodiment of this application;

[0107] FIG. 4 is a flowchart of a method for detecting a second trigger event of a target application by a system service according to an embodiment of this application;

[0108] FIG. 5 is a schematic diagram of generating a shortcut menu after a target application is touched according to an embodiment of this application;

[0109] FIG. 6 is a schematic diagram of a first trigger event occurring on a target application according to another embodiment of this application;

[0110] FIG. 7 is a flowchart of a method for detecting a first trigger event of a target application by a system service according to another embodiment of this application;

[0111] FIG. 8 is a flowchart of a method for detecting a second trigger event of a target application by a system service according to an embodiment of this application; and

[0112] FIG. 9 is a schematic diagram of a structure of an electronic device according to an embodiment of this application.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

[0113] The terms “first” and “second” mentioned below are merely intended for a purpose of description, and shall not be understood as an indication or implication of relative importance or implicit indication of a quantity of indicated technical features. Therefore, a feature limited by “first” or “second” may explicitly indicate or implicitly include one or more such features. In embodiments of this application, the word “example”, “for example”, or the like is used to represent giving an example, an illustration, or a description. Any embodiment or design scheme described as an “example” or “for example” in embodiments of this application should not be explained as being more preferred or having more advantages than another embodiment or design scheme. Exactly, use of the word “example”, “for example”, or the like is intended to present a related concept in a specific manner.

[0114] Unless otherwise defined, all technical and scientific terms used in this specification have same meanings as those usually understood by a person skilled in the art of this application. Terms used in this specification of this application are merely intended to describe specific embodiments, but not to limit this application. It should be understood that “/” indicates “or”, unless otherwise specified. For example, A/B may indicate A or B. The term “and/or” in this specification describes only an association relationship for describing associated objects and represents that three relationships may exist. For example, A and/or B may represent the following three cases: Only A exists, both A and B exist, and only B exists. “At least one” means one or more. “A plurality of” refers to two or more. For example, at least one of a, b, or c may indicate seven cases: a, b, c, a and b, a and c, b and c, and a, b, and c.

[0115] FIG. 1 is a flowchart of a method for accelerating application startup according to an embodiment of this application. The method is applied to an electronic device 1, and may specifically include the following steps.

[0116] Step S101: A system service detects a first trigger event of a target application.

[0117] In this embodiment, when detecting a touch operation on the target application, the system service determines that the first trigger event occurs on the target application. For a detailed process of detecting the first trigger event of the target application by the system service process, refer to FIG. 3, FIG. 6, and the following detailed descriptions of FIG. 3 and FIG. 6.

[0118] Step S102: The system service loads an application program of the target application in response to the first trigger event, and transmits a communication address to the application program.

[0119] In this embodiment, the system service generates a preload instruction in response to the first trigger event, generates the communication address and loads the application program of the target application according to the preload instruction, and transmits the communication address to the application program.

[0120] In this embodiment, after the system service loads the application program of the target application and transmits the communication address to the application program, a mechanism in which the loaded application program communicates with the system service by using a socket is implemented.

[0121] Step S103: The application program performs preloading.

[0122] In this embodiment, the application program loads data of the target application. Specifically, the data, loaded by the application program, of the target application includes application code of the target application and resources of the target application. The application code of the target application includes at least application program package code. The resources of the target application include at least a sound, an image, a font, a cursor, a menu, a dialog box, a toolbar, a handle, and a memory.

[0123] In this embodiment, when detecting the first trigger event of the target application, the system service further determines whether the system service meets a preset condition, and performs preloading in response to the first trigger event when determining that the system service meets the preset condition. Specifically, the system service detects whether remaining use space of the memory is within a preset storage space range, detects whether CPU usage is within a second preset range, detects whether the target application is in a preset application list, and performing preloading in response to the first trigger event when determining that the remaining use space of the memory is within the preset storage space range, the CPU usage is within the second preset range, and the target application is in the preset application list. In this embodiment, the preset target application is an application frequently used by a user. In this way, in this application, when a first trigger event of the application frequently used by the user is detected, the application is preloaded, to increase an application startup rate.

[0124] It should be noted that, when detecting that the remaining use space of the memory is excessively small and is not within the preset storage space range, the system service releases, by using a system memory recycling mechanism, memory resources consumed by another application during preloading. In this way, when it is detected that memory resources are insufficient before the target application performs a preloading process, the memory resources consumed by the another application during preloading are released by using the system memory recycling mechanism, thereby ensuring execution of preloading of the target application.

[0125] Step S104: The application program waits to receive a message based on the communication address.

[0126] In this embodiment, that the application program waits to receive a message based on the communication address includes: the application program creates a server or a client based on the communication address; and the server or the client waits to receive a message based on the communication address. In this embodiment, the communication address is a socket address, the server is a socket server, and the client is a socket server. In this embodiment, the application program creates the server or the client based on the communication address by using a `socket.socket()` function.

[0127] Step S105: The system service detects whether a second trigger event occurs on the target application, where step S106 is performed if the second trigger event occurs on the target application; otherwise, or step S105 is repeatedly performed if no second trigger event occurs on the target application.

[0128] In this embodiment, when detecting the first trigger event of the target application and then releasing the trigger

operation, the system service determines that the second trigger event occurs on the target application. For a detailed process of detecting the second trigger event of the target application by the system service process, refer to FIG. 4, FIG. 7, and the following detailed descriptions of FIG. 4 and FIG. 7.

[0129] Step S106: The system service generates a startup instruction in response to the second trigger event of the target application, and sends the startup instruction to the application program.

[0130] Step S107: The application program receives the startup instruction by using the communication address.

[0131] In this embodiment, the system service sends the startup instruction to the application program by using a socket message. The server or the client in the application program listens to and receives, by using an `accept()` function and the communication address, the startup instruction sent by the system service.

[0132] Step S108: The application program starts the target application in response to the startup instruction.

[0133] In this embodiment, that the application program starts the target application in response to the startup instruction includes: The application program starts the target application by using the data of the target application in response to the startup instruction. In this embodiment, when the application program detects, by using the server or the client, the startup instruction sent by the system service, the application program starts the target application by using the data of the target application in response to the startup instruction, so that socket blocking is implemented when the application program starts the target application. Therefore, in a preloading process of the application program of the target application before the application program of the target application receives the startup instruction, no excessive CPU resources, input/output resources, or memory space needs to be occupied, thereby ensuring controllable preloading behavior of the target application and increasing an application startup speed.

[0134] In this embodiment, after receiving the startup instruction sent by the system service, the application program invokes and executes the application code of the target application by using an `oncreat()` function.

[0135] In this embodiment, before the application program receives the startup instruction by using the server or the client, the method further includes: Before the server or the client receives the startup instruction, the application program suspends execution of the application code of the target application, and waits to receive the startup instruction; the application program skips creating a child thread service; and the application program skips initializing binder communication. In this embodiment, before receiving the startup instruction, the application program suspends execution of the application code of the target application, skips creating a child thread service, and skips initializing binder communication. Therefore, in the preloading process of the application program of the target application before the application program of the target application receives the startup instruction, no CPU resources or input/output resources need to be occupied.

[0136] In this embodiment, when detecting that the user performs the touch operation on the target application on a display screen of the electronic device 1, the system service determines that the first trigger event occurs on the target application. Specifically, FIG. 2A is a schematic diagram of

a first trigger event occurring on a target application according to an embodiment of this application. FIG. 3 is a flowchart of a method for detecting a first trigger event of a target application by a system service according to an embodiment of this application. The method may specifically include the following steps.

[0137] Step S301: In response to the touch operation, the system service determines a touch location corresponding to the touch operation.

[0138] Step S302: The system service determines, based on the touch location, whether the target application is touched.

[0139] Step S303: If it is determined that the target application is touched, the system service determines that the first trigger event occurs on the target application.

[0140] In an embodiment, when detecting that the user performs the touch operation on the target application and then stops performing the touch operation on the target application, the system service determines that the second trigger event occurs on the target application. Specifically, FIG. 2B is a schematic diagram of a second trigger event occurring on a target application according to an embodiment of this application. FIG. 4 is a flowchart of a method for detecting a second trigger event of a target application by a system service according to an embodiment of this application. The method may specifically include the following steps.

[0141] Step S401: When determining that the target application is touched, the system service performs timing to obtain a timing time.

[0142] Step S402: Determine whether the touch operation on the target application ends within a first preset time range, where step S403 is performed if the touch operation on the target application ends within the first preset time range, or step S404 is performed if the touch operation on the target application does not end within the first preset time range.

[0143] Step S403: Determine that the second trigger event occurs on the target application.

[0144] Step S404: Determine whether the target application ends within a second preset time range, where step S405 is performed if the touch operation ends within the second preset time range, or step S406 is performed if the touch operation does not end within the second preset time range.

[0145] Step S405: The system service sleeps for a preset period of time.

[0146] Step S406: Determine that a third trigger event occurs on the target application, and display a shortcut menu of the target application in response to the third trigger event.

[0147] Step S402: Determine whether the touch operation ends, where step S403 is performed if the touch operation, or step S404 is performed if the touch operation does not end.

[0148] FIG. 5 is a schematic diagram of generating a shortcut menu after a target application is touched according to an embodiment of this application. In this embodiment, when the system service determines that the third trigger event occurs on the target application, the system service does not start the target application, but displays the shortcut menu of the target application for the user to perform an operation on a corresponding shortcut function of the target application. As shown in FIG. 5, when a time of touching and holding a camera application by the user exceeds the second preset time range, the system service does not start the target application, but displays the shortcut menu of the

target application. The displayed shortcut menu includes at least a selfie function option, and a selfie function of the camera application is performed when the user selects the selfie function option. In an embodiment, when detecting a first speech instruction for the target application, the system service determines that the first trigger event occurs on the target application. Specifically, FIG. 6 is a schematic diagram of a first trigger event occurring on a target application according to an embodiment of this application. FIG. 7 is a flowchart of a method for detecting a first trigger event of a target application by a system service according to an embodiment of this application. The method may specifically include the following steps.

[0149] Step S701: The system service obtains the first speech instruction.

[0150] Step S702: Analyze whether an operation object of the first speech instruction is the target application.

[0151] Step S703: If the operation object of the first speech instruction is the target application, determine whether the first speech instruction includes a first control instruction.

[0152] Step S704: If it is determined that the first speech instruction includes the first control instruction, determine that the first trigger event occurs on the target application.

[0153] In this embodiment, the system service recognizes the first speech instruction by using a speech recognition unit. The first speech instruction includes the operation object and the first control instruction for the operation object. For example, the first speech instruction may be “load a music player”, where the “music player” is the operation object, and “load” is the first control instruction. If the system service detects the first speech instruction “load a music player”, the system service determines that the target application in the first speech instruction is the “music player”, determines that the first control instruction in the first speech instruction is “load”, and determines that the first trigger event occurs on the “music player”.

[0154] In an embodiment, when detecting a second speech instruction for the target application, the system service determines that the second trigger event occurs on the target application. Specifically, FIG. 8 is a flowchart of a method for detecting a second trigger event of a target application by a system service according to an embodiment of this application. The method may specifically include the following steps.

[0155] Step S801: The system service obtains the second speech instruction.

[0156] Step S802: Analyze whether an operation object of the second speech instruction is the target application.

[0157] Step S803: If the operation object of the second speech instruction is the target application, determine whether the second speech instruction includes a second control instruction.

[0158] Step S804: If it is determined that the second speech instruction includes the second control instruction, determine that the second trigger event occurs on the target application.

[0159] In this embodiment, the second speech instruction includes the operation object and the second control instruction for the operation object. For example, the second speech instruction may be “start a music player”, where the “music player” is the operation object, and “start” is the second control instruction. If the system service detects the second speech instruction “start a music player”, the system service

determines that the target application in the second speech instruction is the “music player”, determines that the second control instruction in the second speech instruction is “start”, and determines that the second trigger event occurs on the “music player”.

[0160] It should be noted that, when the system service detects the first speech instruction for the target application and determines that the first trigger event occurs on the target application, the method is not limited to an implementation of determining, by detecting a speech instruction, that the second trigger event occurs on the target application, but may further include an implementation of determining, by detecting that the user performs a touch operation on the target application and then stops the touch operation on the target application, that the second trigger event occurs on the target application.

[0161] In an embodiment, the system service detects eyeball tracking information of the user, and determines, based on the eyeball tracking information, the target application and whether the first trigger event occurs on the target application. Specifically, the system service obtains the eyeball tracking information of the user, and determines, based on the eyeball tracking information, that the first trigger event occurs on a target application whose gaze duration exceeds first preset duration. In this embodiment, the system service may obtain the tracking information by using an eyeball tracking apparatus. The eyeball tracking apparatus is an infrared sensor or a camera. Longer duration of gazing at an application by the user indicates that the user is more likely to start the application program. Therefore, an application corresponding to an application icon on which gaze duration of eyes of the user is longer than preset duration may be determined as the target application based on the eyeball tracking information. In another embodiment, the system service determines an eyeball motion trajectory of the user based on the eyeball tracking information; determines, based on the eyeball motion trajectory, a quantity of times that the user looks at each application icon on the electronic device 1; and determines that the first trigger event occurs on a target application for which the quantity of times exceeds a first preset threshold.

[0162] In an embodiment, the system service further determines, based on the eyeball tracking information, whether the second trigger event occurs on the target application. Specifically, the system service determines, based on the eyeball tracking information, whether gaze duration of the target application exceeds second preset duration; and if the gaze duration of the target application exceeds the second preset duration, determines that the second trigger event occurs on the target application. In another embodiment, the system service determines, based on the eyeball tracking information, a quantity of times that an eyeball motion trajectory of the user passes an icon of the target application; and when the quantity of times exceeds a second preset threshold, determines that the first trigger event occurs on the target application.

[0163] In an embodiment, the system service determines, based on location information of the electronic device 1, whether the first trigger event occurs on the target application. Specifically, the system service obtains a location of the electronic device 1; determines whether the location of the electronic device 1 is within a preset area range; and if it is determined that the location of the electronic device 1 is within the preset area range, determines that the first trigger

event occurs on the target application on the electronic device 1. In another embodiment, the system service determines, based on time information, whether the first trigger event occurs on the target application. Specifically, the system service determines whether a current time is within a preset time range; and if it is determined that the current time is within the preset time range, determines that the first trigger event occurs on the target application. In this embodiment, the target application is an application of a preset type. Preset types include a game application type and a video application type. It should be noted that start time of a game application and start time of a video application are long. Therefore, the user may set the two types as the preset types, so that application programs of the two types can be preloaded, to reduce a waiting time during startup of the application programs.

[0164] In an embodiment, the system service performs timing after determining that the first trigger event occurs on the target application, and determines, based on a timing time, whether the second trigger event occurs on the target application. Specifically, when determining that the first trigger event occurs on the target application, the system service performs timing to obtain the timing time; the system service determines whether the timing time is greater than a second preset time threshold; and if the system service determines that the timing time is greater than the second preset time threshold, it is determined that the second trigger event occurs on the target application.

[0165] In an embodiment, the system service analyzes, by using a machine learning algorithm, a behavior condition of using the target application by the user; and if it is determined that the target application meets the behavior condition, determines that the first trigger event occurs on the target application. Specifically, the system service obtains first historical data of using the target application by the user, where the first historical data includes at least time information, frequency information, and location information of using the target application; the system service analyzes the first historical data by using a machine learning method, and determines the behavior condition of the user for the target application; the system service determines whether the target application meets the behavior condition; and if it is determined that the target application meets the behavior condition, it is determined that the first trigger event occurs on the target application.

[0166] For example, the system service analyzes, by using the machine learning method, the time information of using the target application to determine a time period in which the user uses the target application, and uses the time period in which the user uses the target application as the behavior condition. When determining that a current time is within the time period in which the target application is used, the system service determines that the target application meets the behavior condition, and determines that the first trigger event occurs on the target application.

[0167] For another example, the system service analyzes, by using the machine learning method, the frequency information of using the target application to determine frequency of using the target application by the user, and uses the frequency of using the target application by the user as the behavior condition. When determining that the frequency of using the target application exceeds a preset frequency threshold, the system service determines that the

target application meets the behavior condition, and determines that the first trigger event occurs on the target application.

[0168] For another example, the system service analyzes, by using the machine learning method, the location information of using the target application to determine a location at which the user uses the target application, and uses the location at which the user uses the target application as the behavior condition. When determining that the location at which the target application is used is within a preset location range, the system service determines that the target application meets the behavior condition, and determines that the first trigger event occurs on the target application.

[0169] In another embodiment, the system service may use, as the behavior condition, a combination of the time period in which the user uses the target application, the frequency of using the target application by the user, and the location at which the user uses the target application; and when determining that the target application meets the behavior condition, determine that the first trigger event occurs on the target application. For the determining whether the target application meets the behavior condition, refer to the descriptions in the foregoing content. Details are not described herein again.

[0170] In an embodiment, when receiving a preset broadcast message, the system service determines that the second trigger event occurs on the target application. The preset broadcast message indicates to start a specified target application.

[0171] In an embodiment, when receiving a first preset instruction sent by a service component, the system service determines that the second trigger event occurs on the target application. The first preset instruction instructs to start a specified target application.

[0172] In an embodiment, when receiving a first preset instruction sent by a service component, the system service determines that the second trigger event occurs on the target application. The first preset instruction instructs to start a specified target application.

[0173] In an embodiment, when receiving a second preset instruction sent by an activity component, the system service determines that the second trigger event occurs on the target application. The second preset instruction instructs to start a specified target application.

[0174] In an embodiment, when receiving a third preset instruction sent by a provider component, the system service determines that the second trigger event occurs on the target application. The third preset instruction instructs to start a specified target application.

[0175] In an embodiment, that the system service generates a preload instruction in response to the first trigger event, to preload the target application includes: when the system service detects that the first trigger event occurs on a plurality of target applications, obtaining second historical data of the plurality of target applications; determining a high-frequency usage time period or usage frequency of the plurality of target applications based on the second historical data of the plurality of target applications; sorting the plurality of target applications based on a high-frequency usage time period or usage frequency of each target application; and sequentially preloading the plurality of application programs based on the sorting. In this way, when the first trigger event occurs on the plurality of target applications, the system service may preload a target application

program with a higher level based on priority levels, thereby facilitating usage by the user.

[0176] FIG. 8 is a schematic diagram of a structure of an electronic device 1 according to an embodiment of this application. The electronic device 1 includes but is not limited to an eyeball tracking apparatus 11, a speech recognition unit 12, a display screen 13, a processor 14, and a memory 15. The foregoing devices may be connected by using one or more communication buses 16. The eyeball tracking apparatus 11 is an infrared sensor or a camera. The speech recognition unit 12 is a speech recognition area. The display screen 13 is a liquid crystal display with a touch function. The memory 15 is configured to store one or more computer programs 17. The one or more computer programs 17 are configured to be executed by the processor 14. The one or more computer programs 17 include a plurality of instructions. When the plurality of instructions are executed by the processor 14, the method for accelerating application startup that is performed on the electronic device 1 in the foregoing embodiments may be implemented, to implement a function, of the electronic device 1, of accelerating application startup.

[0177] An embodiment further provides a computer storage medium. The computer storage medium stores computer instructions. When the computer instructions are run on an electronic device, the electronic device is enabled to perform the foregoing related method steps to implement the unlock control method in the foregoing embodiments.

[0178] An embodiment further provides a computer program product. When the computer program product is run on a computer, the computer is enabled to perform the foregoing related steps to implement the unlock control method in the foregoing embodiments.

[0179] In addition, an embodiment of this application further provides an apparatus. The apparatus may be specifically a chip, a component, or a module, and the apparatus may include a processor and a memory that are connected to each other. The memory is configured to store computer-executable instructions. When the apparatus runs, the processor may execute the computer-executable instructions stored in the memory, so that the chip performs the unlock control method in the foregoing method embodiments.

[0180] The electronic device, the computer storage medium, the computer program product, and the chip provided in the embodiments are all configured to perform the corresponding methods provided in the foregoing descriptions. Therefore, for beneficial effects that can be achieved by the electronic device, the computer storage medium, the computer program product, and the chip, refer to the beneficial effects of the corresponding methods provided in the foregoing descriptions. Details are not described herein again.

[0181] Based on the foregoing descriptions of the implementations, a person skilled in the art may clearly understand that, for the purpose of convenient and brief descriptions, division into the foregoing functional modules is merely used as an example for descriptions. During actual application, the foregoing functions can be allocated to different functional modules for implementation based on a requirement, in other words, an inner structure of an apparatus is divided into different functional modules to implement all or some of the functions described above.

[0182] In the several embodiments provided in this application, it should be understood that the disclosed apparatus

tuses and methods may be implemented in other manners. For example, the described apparatus embodiments are merely examples. For example, division into the modules or units is merely logical function division, and there may be other division manners during actual implementation. For example, a plurality of units or components may be combined or may be integrated into another apparatus, or some features may be ignored or not performed. In addition, the displayed or discussed mutual couplings or direct couplings or communication connections may be implemented through some interfaces. The indirect couplings or communication connections between the apparatuses or units may be implemented in electrical, mechanical, or other forms.

[0183] The units described as separate parts may or may not be physically separate, and parts displayed as units may be one or more physical units, that is, may be located in one place, or may be distributed in a plurality of different places. Some or all of the units may be selected based on actual requirements to achieve the objectives of the solutions of embodiments.

[0184] In addition, functional units in embodiments of this application may be integrated into one processing unit, each of the units may exist alone physically, or two or more units may be integrated into one unit. The integrated unit may be implemented in a form of hardware, or may be implemented in a form of a software functional unit.

[0185] When the integrated unit is implemented in a form of a software functional unit and sold or used as an independent product, the integrated unit may be stored in a readable storage medium. Based on such an understanding, the technical solutions of embodiments of this application essentially, or the part contributing to the conventional technology, or all or some of the technical solutions may be implemented in a form of a software product. The software product is stored in a storage medium and includes several instructions for instructing a device (which may be a single-chip microcomputer, a chip, or the like) or a processor (processor) to perform all or some of steps of methods in embodiments of this application. The foregoing storage medium includes: any medium that can store program code, such as a USB flash drive, a removable hard disk, a read-only memory (Read-Only Memory, ROM), a random access memory (Random Access Memory, RAM), a magnetic disk, or an optical disc.

[0186] Finally, it should be noted that the foregoing embodiments are merely intended to describe the technical solutions of this application, but should not be construed as a limitation. Although this application is described in detail with reference to example embodiments, a person of ordinary skill in the art should understand that modifications or equivalent replacements may be made to the technical solutions of this application, without departing from the spirit and scope of the technical solutions of this application.

1. A method, wherein the method comprises:

detecting, by a system service, a first trigger event of a target application;

loading, by the system service, an application program of the target application in response to the first trigger event, and transmitting a communication address to the application program;

performing, by the application program, preloading;

waiting, by the application program, to receive a message based on the communication address;

detecting, by the system service, a second trigger event of the target application;

generating, by the system service, a startup instruction in response to detecting the second trigger event of the target application, and sending the startup instruction to the application program;

receiving, by the application program, the startup instruction by using the communication address; and

starting, by the application program, the target application in response to receiving the startup instruction.

2. The method according to claim **1**, wherein waiting, by the application program, to receive the message based on the communication address comprises:

creating, by the application program, a server or a client based on the communication address; and

waiting, by the server or the client, to receive a message based on the communication address.

3.-7. (canceled)

8. The method according to claim **1**, wherein detecting, by the system service, the first trigger event of the target application comprises:

when detecting a touch operation on the target application, determining, by the system service, that the first trigger event occurs on the target application.

9. The method according to claim **8**, wherein determining, by the system service when detecting the touch operation on the target application, that the first trigger event occurs on the target application comprises:

in response to detecting the touch operation, determining, by the system service, a touch location corresponding to the touch operation;

determining, based on the touch location, whether the target application is touched; and

when it is determined that the target application is touched, determining that the first trigger event occurs on the target application.

10. The method according to claim **8**, wherein detecting, by the system service, the second trigger event of the target application comprises:

after detecting the first trigger event of the target application and determining that the touch operation corresponding to the first trigger event is released, determining, by the system service, that the second trigger event occurs on the target application.

11. The method according to claim **10**, wherein detecting, by the system service, the second trigger event of the target application comprises:

when determining that the target application is touched, performing, by the system service, timing to obtain a timing time;

determining whether the touch operation on the target application ends within a first preset time range; and

in response to determining the touch operation on the target application ends within the first preset time range, determining that the second trigger event occurs on the target application.

12. The method according to claim **11**, wherein the method further comprises:

in response to determining the touch operation on the target application does not end within the first preset time range, determining, by the system service, whether the touch operation on the target application ends within a second preset time range; and

- in response to determining the touch operation ends within the second preset time range, sleeping, by the system service, for a preset period of time; or
- in response to determining the touch operation does not end within the second preset time range, determining, by the system service, that a third trigger event occurs on the target application, and displaying a shortcut menu of the target application in response to the third trigger event.
- 13.** The method according to claim 1, wherein detecting, by the system service, the first trigger event of the target application comprises:
- obtaining, by the system service, a first speech instruction;
 - analyzing whether an operation object of the first speech instruction is the target application;
 - when the operation object of the first speech instruction is the target application, determining whether the first speech instruction comprises a first control instruction; and
 - when it is determined that the first speech instruction comprises the first control instruction, determining that the first trigger event occurs on the target application.
- 14.** The method according to claim 1, wherein detecting, by the system service, the second trigger event of the target application comprises:
- obtaining, by the system service, a second speech instruction;
 - analyzing whether an operation object of the second speech instruction is the target application;
 - when the operation object of the second speech instruction is the target application, determining whether the second speech instruction comprises a second control instruction; and
 - when it is determined that the second speech instruction comprises the second control instruction, determining that the second trigger event occurs on the target application.
- 15.** The method for accelerating application startup according to claim 1, wherein detecting, by the system service, the first trigger event of the target application comprises:
- obtaining, by the system service, eyeball tracking information of a user; and
 - determining, based on the eyeball tracking information, that the first trigger event occurs on a target application whose gaze duration exceeds a first preset duration.
- 16.** The method according to claim 15, wherein detecting, by the system service, the second trigger event of the target application comprises:
- determining, based on the eyeball tracking information, whether a gaze duration of the target application exceeds a second preset duration; and
 - in response to determining that the gaze duration of the target application exceeds the second preset duration, determining that the second trigger event occurs on the target application.
- 17.** The method according to claim 1, wherein detecting, by system service, the first trigger event of the target application comprises:
- obtaining, by the system service, a location of an electronic device;
 - determining whether the location of the electronic device is within a preset area range; and
- when the location of the electronic device is within the preset area range, determining that the first trigger event occurs on the target application of the electronic device.
- 18.** The method according to claim 1, wherein detecting, by the system service, the first trigger event of the target application comprises:
- determining, by the system service, whether a current time is within a preset time range; and
 - when it is determined that the current time is within the preset time range, determining that the first trigger event occurs on the target application.
- 19.** The method according to claim 17, wherein detecting, by the system service, the second trigger event of the target application comprises:
- when determining that the first trigger event occurs on the target application, performing, by the system service, timing to obtain a timing time;
 - determining, by the system service, whether the timing time is greater than a second preset time threshold; and
 - in response to determining that the system service determines that the timing time is greater than the second preset time threshold, determining that the second trigger event occurs on the target application.
- 20.** The method according to claim 17, wherein the target application is an application of a preset type.
- 21.** The method according to claim 1, wherein detecting, by a system service, the first trigger event of a target application comprises:
- obtaining, by the system service, first historical data of using the target application by a user, wherein the first historical data comprises at least time information, frequency information, and location information of using the target application;
 - analyzing, by the system service, the first historical data by using a machine learning method, and determining a usage habit condition of the user for the target application;
 - determining, by the system service, whether the target application meets the usage habit condition; and
 - when it is determined that the target application meets the usage habit condition, determining that the first trigger event occurs on the target application.
- 22.** The method according to claim 1, wherein loading, by the system service, the application program of the target application in response to the first trigger event comprises:
- when the system service detects that the target application comprises a plurality of target applications, obtaining second historical data of the plurality of target applications;
 - determining a high-frequency usage time period or usage frequency of the plurality of target applications based on the second historical data of the plurality of target applications;
 - sorting the plurality of target applications based on a high-frequency usage time period or usage frequency of each target application; and
 - sequentially preloading the plurality of target applications according to the sorting.
- 23.** An electronic device, wherein the electronic device comprises a memory and a processor, wherein
- the memory is configured to store program instructions; and

the processor is configured to read and execute the program instructions stored in the memory, and when the program instructions are executed by the processor, the electronic device is enabled to perform:

detecting, by a system service of the electronic device, a first trigger event of a target application of the electronic device;

loading, by the system service, an application program of the target application in response to the first trigger event, and transmitting a communication address to the application program;

performing, by the application program, preloading;

waiting, by the application program, to receive a message based on the communication address;

detecting, by the system service, a second trigger event of the target application;

generating, by the system service, a startup instruction in response to the second trigger event of the target application, and sending the startup instruction to the application program;

receiving, by the application program, the startup instruction by using the communication address; and

starting, by the application program, the target application in response to the startup instruction.

24. A computer storage medium, wherein the computer storage medium stores program instructions, and when the program instructions are run on an electronic device, the electronic device is enabled to perform:

detecting, by a system service of the electronic device, a first trigger event of a target application of the electronic device;

loading, by the system service, an application program of the target application in response to the first trigger event, and transmitting a communication address to the application program;

performing, by the application program, preloading;

waiting, by the application program, to receive a message based on the communication address;

detecting, by the system service, a second trigger event of the target application;

generating, by the system service, a startup instruction in response to the second trigger event of the target application, and sending the startup instruction to the application program;

receiving, by the application program, the startup instruction by using the communication address; and

starting, by the application program, the target application in response to the startup instruction.

* * * * *