

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 11/07 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200680017717.0

[43] 公开日 2008年9月17日

[11] 公开号 CN 101268447A

[22] 申请日 2006.5.19

[21] 申请号 200680017717.0

[30] 优先权

[32] 2005.5.26 [33] US [31] 11/139,339

[32] 2005.5.26 [33] US [31] 11/139,340

[32] 2005.5.26 [33] US [31] 11/139,341

[86] 国际申请 PCT/US2006/019529 2006.5.19

[87] 国际公布 WO2006/127493 英 2006.11.30

[85] 进入国家阶段日期 2007.11.22

[71] 申请人 美国联合包裹服务公司

地址 美国佐治亚州

[72] 发明人 朱莲·约克·亚伯

[74] 专利代理机构 广州三环专利代理有限公司
代理人 郝传鑫

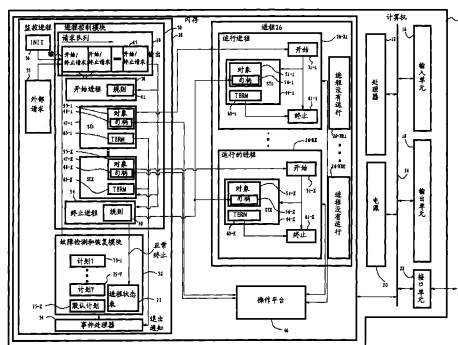
权利要求书 6 页 说明书 14 页 附图 5 页

[54] 发明名称

软件进程监控

[57] 摘要

监控进程在启动计算机后执行一系列客户端进程，开始，终止，监控运行于计算机上的进程，和/或自动检测和恢复进程故障。当客户端进程被启动时，进程使用它创建的监控线程监控客户端进程，并使用进程的句柄检测进程终止。当检测到进程终止，检查监控进程内的状态表以确定是否是客户端进程故障或是正常终止。如果是由于故障导致终止，进程使用预先配置的计划自动执行故障恢复。



1、一种计算机，其特征在于：包括

存储有监控进程的内存，所述内存包括：

进程控制模块，其基于对客户端进程外部定义并存储在所述内存中的规则开始或终止客户端进程，所述进程控制模块通过请求操作平台启动客户端进程以启动所述客户端进程，从所述操作平台接收所述已启动客户端进程的句柄，生成第一监控线程并将其与句柄并入到所述已启动客户端进程中，并储存所述第一监控线程在所述进程控制模块中，用于监控客户端进程的运行状态，所述进程控制模块产生退出通知事件以响应通过所述监控线程对已终止客户端进程的检测，所述进程控制模块通过在客户端进程中发信号给第二监控线程的进程间的通信对象，利用进程间的通信对象的句柄，终止所述客户端进程；

故障检测和恢复模块，其为一个或多个启动的客户端进程存储定义的故障恢复计划，所述故障检测和恢复模块具有进程状态表，如果客户端进程已经正常终止，则从所述进程控制模块接收正常终止信号，并从已终止进程经由相对应于所述已终止客户端进程的所述第一监控线程接收退出通知信号，所述退出通知信号表明客户端进程终止，所述故障检测和恢复模块检查以决定是否进程是正常终止，以响应从所述进程控制模块接收的退出通知信号，所述检查是检查终止的客户端进程是否有对应的正常终止数据在进程状态表中，所述故障检测和恢复模块存储至少一个定义计算机怎样从客户端进程故障恢复的故障恢复计划，如果所述进程控制模块确定客户端进程是由于故障导致退出，则所述进程控制模块执行故障恢复计划；

以及

处理器，其连接到所述内存，并控制所述进程控制模块和故障检测模块执行所述功能。

2、如权利要求1所述的计算机，其特征在于，所述内存进一步存储一个事件处理器，所述事件处理器从所述进程控制模块接收退出通知信号，所述退出通知信号表示客户端进程已经终止，所述事件处理器将所述退出通知信号通知所述故障检测和恢复模块，所述故障检测和恢复模块确定客户端进程是否正常

终止以响应来自所述进程控制模块的对应的退出通知信号。

3、如权利要求1所述的计算机，其特征在于，所述进程控制模块含有一个队列，所述队列存储由所述进程控制模块在规则允许下执行的开始和终止请求。

4、如权利要求3所述的计算机，其特征在于，所述监控进程包含一个经配置以定义在计算机启动时被执行的开始和终止请求的初始化进程，所述初始化进程输入所述经配置的开始和终止请求到所述队列里以在计算机启动时执行。

5、如权利要求1所述的计算机，其特征在于，至少一个开始和终止请求输入到运行进程的队列中。

6、一种计算机，其特征在于，包括：

处理器；以及

所述处理器连接的内存；

为运行在所述计算机上的对应的客户端进程，所述内存存储一个含有监控线程的监控进程，所述监控线程外部运行对应的客户端进程，允许所述处理器外部监控对应的客户端进程。

7、如权利要求6所述的计算机，其特征在于，所述监控进程包含一个开始进程，所述开始进程使用来自操作平台的句柄以响应运行在所述操作平台的客户端进程的开始，并且创建监控线程，所述监控线程包括用于监控所述客户端进程的句柄。

8、如权利要求7所述的计算机，其特征在于，所述开始进程包括一个或多个规则，所述规则用于开始进程确定客户端进程是否可被启动，所述开始进程启动所述客户端进程开始以响应确定基于对客户端进程外部已定义的一个或多个规则的客户端进程可被启动。

9、如权利要求6所述的计算机，其特征在于，还包括：

输入单元；

输出单元；以及

接口单元；

所述监控进程包括开始进程和/或终止进程，其均能被另一个由处理器执行的运行进程触发，使用者通过所述输入单元和所述输出单元，外部设备通过所述接口单元，或者它们的结合，使用监控线程、一个或多个对应客户端进程的句柄、在对应的客户端进程内进程间通信对象的句柄，来控制所述客户端进程。

10、如权利要求 6 所述的计算机，其特征在于，所述内存存储一个初始化进程，所述初始化进程定义了一个或多个由处理器在计算机开启时启动的进程。

11、如权利要求 6 所述的计算机，其特征在于，所述监控进程进一步包括一个故障检测和恢复模块，所述故障检测和恢复模块存储了一个定义从所述客户端进程的故障恢复的故障恢复计划，所述处理器执行所述恢复计划以在没有包含计算机使用者请求的情形下从所述客户端进程的故障自动恢复。

12、如权利要求 11 所述的计算机，其特征在于，所述故障检测和恢复模块包括存储正常终止标识数据的进程状态表，所述正常终止标识数据是由监控进程的终止进程所设置以表示客户端进程正常终止。

13、如权利要求 11 所述的计算机，其特征在于，所述监控进程进一步包括一个事件处理器，在所述客户端进程终止时，所述事件处理器从所述客户端进程的监控线程接收一个退出通知，并激活所述故障检测和恢复模块用以确定客户端进程是否正常终止或是由于故障而终止。

14、一种计算机可读介质，其特征在于，所述计算机可读介质存储一个计算机进程，计算机执行所述计算机进程的步骤包括：

给客户端进程创建一个外部监控线程用以启动运行在计算机处理器上的所述客户端进程，所述创建的监控线程包括一个从操作平台接收的句柄，所述句柄被所述处理器用来监所述控客户端进程的运行状态。

15、如权利要求 14 所述的计算机可读介质，其特征在于，所述计算机执行所述计算机进程的步骤进一步包括：

由处理器执行一个初始化进程以响应启动，重新启动进程，或者重新启动计算机，或者它们的结合，所述初始化进程执行启动一个或多个由所述初始化进程定义的客户端进程，创建为每一各自由所述初始化进程定义的客户端进程执行的监控线程，以使每一个运行的客户端进程具有对应的监控线程，所述处理器使用所述监控线程监控每一个运行进程的状态。

16、如权利要求 14 所述的计算机可读介质，其特征在于，所述计算机执行所述计算机进程的步骤进一步包括：

接收请求以启动客户端进程，创建响应于接收的请求而执行的监控线程。

17、如权利要求 16 所述的计算机可读介质，其特征在于，所述计算机执行所述计算机进程的步骤进一步包括：

响应接收请求，确定所述客户端进程基于一个或多个外部定义的规则是否被启动，执行启动以响应确定客户端进程基于一个或多个规则被启动。

18、如权利要求 16 所述的计算机可读介质，其特征在于，所述请求从另一个由所述处理器执行的进程接收。

19、如权利要求 16 所述的计算机可读介质，其特征在于，所述请求通过计算机输入单元从使用者接收。

20、如权利要求 16 所述的计算机可读介质，其特征在于，所述请求通过计算机接口单元从外部设备接收。

21、如权利要求 14 所述的计算机可读介质，其特征在于，所述计算机执行所述计算机进程的步骤进一步包括：

在所述操作平台启动所述客户端进程后，从所述操作平台接收句柄，接收所述句柄在创建所述监控线程之前。

22、一种计算机可读介质，存储一个计算机进程，计算机执行所述计算机进程的步骤包括：创建客户端进程的内部监控线程以响应启动运行在计算机处理器上的客户端进程，所述监控线程由来自监控进程的终止请求激活，作为在为发信号给客户端进程以终止执行的进程中的目标，所述监控进程传送所述终止请求给所述客户端进程，使用所述进程的监控线程的进程间通信对象的句柄使所述监控进程与所述客户端进程通信。

23、如权利要求 22 所述的计算机可读介质，其特征在于，所述进程间通信对象包括至少一个信号量，互斥，事件，或它们的结合。

24、如权利要求 22 所述的计算机可读介质，其特征在于，所述计算机执行所述计算机进程的步骤进一步包括：

接收请求以在所述监控进程中终止所述客户端进程；以及

通过发信号给所述进程中的进程间通信对象的句柄以响应接收请求，从所述监控进程终止所述客户端进程。

25、如权利要求 24 所述的计算机可读介质，其特征在于，还包括步骤：

响应于接收所述请求，确定客户端进程是否基于一个或多个外部定义客户端进程的规则被终止，执行终止以响应确定进程基于一个或多个规则可被终止。

26、如权利要求 22 所述的计算机可读介质，其特征在于，所述请求从由所述处理器执行的另一个进程接收。

27、如权利要求 22 所述的计算机可读介质，其特征在于，所述请求通过计算机输入单元从使用者接收。

28、如权利要求 22 所述的计算机可读介质，其特征在于，所述请求通过计算机接口单元从外部设备接收。

29、一种计算机，其特征在于：

处理器；以及

连接到所述处理器的内存，所述内存存储有故障检测和恢复模块，所述故障和恢复模块为由所述处理器执行的客户端进程定义一个恢复计划，所述故障检测和恢复模块通过在没有由计算机使用者请求的情况下执行所述恢复计划，自动检测和恢复所述客户端进程故障。

30、如权利要求 29 所述的计算机，其特征在于，所述恢复计划为所述客户端进程特定定义。

31、如权利要求 29 所述的计算机，其特征在于，所述恢复计划是默认的计划，用于在没有为所述客户端进程特定定义恢复计划的事件中恢复所述客户端进程故障。

32、如权利要求 29 所述的计算机，其特征在于，所述恢复计划表示所述计算机被重新启动。

33、如权利要求 29 所述的计算机，其特征在于，所述恢复计划表示所述客户端进程被重新启动。

34、如权利要求 29 所述的计算机，其特征在于，所述恢复计划表示所述处理器终止至少一个其他客户端进程。

35、如权利要求 29 所述的计算机，其特征在于，所述恢复计划表示所述处理器启动至少一个其他客户端进程。

36、一种计算机可读介质，其特征在于，所述计算机可读介质存储一个计算机进程，所述计算机执行所述计算机进程的步骤包括：

所述客户端进程外部检测运行在所述计算机处理器的客户端进程故障；以及

自动执行一个故障恢复计划以恢复所述客户端进程故障。

37、如权利要求 36 所述的计算机可读介质，其特征在于，所述检测步骤进一步包括：

通过终止客户端进程的句柄的无效，发送信号给监控线程，所述监控线程产生退出通知事件以响应所述句柄的无效；

接收所述退出通知事件；

在终止前，检查由所述客户端进程设置的进程终止标识；以及

基于所述检查，确定所述客户端进程是否故障；

基于所述确定，执行所述故障恢复计划。

38、如权利要求 36 所述的计算机可读介质，其特征在于，为客户端进程定义特定定义的所述故障恢复计划。

39、如权利要求 36 所述的计算机可读介质，其特征在于，所述故障恢复计划是在没有为客户端进程特定定义的计划时定义的默认计划。

40、如权利要求 36 所述的计算机可读介质，其特征在于，所述故障恢复计划引导所述处理器重新启动计算机。

41、如权利要求 36 所述的计算机可读介质，其特征在于，所述故障恢复计划引导处理器重新启动发生故障的客户端进程。

42、如权利要求 36 所述的计算机可读介质，其特征在于，所述故障恢复计划引导所述处理器启动至少一个附加的进程。

43、如权利要求 36 所述的计算机可读介质，其特征在于，所述故障恢复计划引导所述处理器终止至少一个附加的进程。

软件进程监控

技术领域

本发明涉及一种用以监控和管理进程，如在计算机上运行的请求进程、交互进程和效用进程的装置以及计算机可读介质。

背景技术

计算机使用进程如请求进程，交互进程，和效用进程实现使用者与计算机之间的直接通信任务。在多数计算机环境中，每一个进程独立运行并定义关于进程开始，终止和其他进程间交互的规则。因此，从一般角度看，这是很难确定管理进程运行、进程和其他进程之间交互的规则，因为这些规则是遮蔽在进程的代码中。最好是能够提供一个管理和控制进程外部运行的系统，管理进程开始、终止和其他进程间交互的规则可以被外部定义。这将大大简化对进程的控制，使管理进程交互的规则以一个相对简单的方式让程序员理解。

这些技能在技术上理解，由于各种原因，进程运行在计算机上会产生'故障'，大致意思是说这个代码不正确操作或以其他方式意外结束。故障是由许多原因引起的，包括，举例来说，在进程中存在错误，进程接收输入状态没有被编程响应，由于错误或其他原因产生数据或代码错误，内存或缓存溢出，代码冲突，及其他。最好能提供一个系统具有外部监控进程检测任何不当进程操作。此外，它还将是可取的，在发生故障时，使进程通过执行重新启动计算机，重新启动进程，或两者兼而有之，自动而不需要任何使用者行为就能够使进程运行。此外，它还将是可取的，如果使用为进程预先配置的恢复计划，进程重新启动或是设备重新启动，提供灵活的方式使进程从故障中恢复。

现有技术的另一个问题是一个进程控制其他进程的方式。在现有技术中，在一个受控制的方式下，一个运行的进程没有能力启动或终止另一个相关进程。最好是能够启动一个进程具有管理能力，这个能力是在稍后时间，使外部代码在特殊情况下终止，在发生故障时，提前配置它的恢复行为。在一个进程需要终止的情况时，最好提供能够在进程终止之前实现配置退出处理。

因此，应该提供一个装置，计算机可读介质，监控和管理客户端进程的方法，如计算机上可运行的请求进程，交互进程，和效用进程，具有克服上述现有技术存在的问题。

发明内容

本发明实施例所要解决的技术问题在于，提供一种计算机，所述计算机包含一个处理器和连接到所述处理器的内存。所述内存存储有监控进程，所述监控进程拥有运行在计算机上与进程对应的监控线程。监控线程单独运行或外部运行对应的进程，从而使处理器能够使用监控线程外部控制其对应的进程。监控进程包括一个开始进程，用于从操作平台接收一个句柄，以响应运行在操作平台上的开始进程，创建包含句柄的监控线程用于监控进程。这个进程可定制允许监控进程与其交互。众所周知，无论是监控进程和对应的进程，一个监控线程可以内部添加到进程中，监控特殊进程间的通信对象（例如信号量，互斥，事件等）。监控进程可以通过发信号给进程间的通信对象请求终止对应的进程，触发进程，执行定制的终止程序终止进程本身。终止进程程序可能涉及存储数据，或在终止之前执行其他任务。可以通过其中的各种方式触发开始进程和终止进程，并可能源自通过处理器执行另一个运行进程，使用者通过使用界面所提供的计算机的输入和输出单元，与计算机连接的外部设备，计算机的接口单元，或综合所述。监控进程基于一个或多个外部规则执行开始和终止请求，即确定是否在此情况下的一个进程，目标请求进程可以被开始或终止。所述内存可以储存一个初始化进程确定一个或多个进程，这些进程在计算机开启后由处理器启动。所述监控进程可进一步构成一个故障检测和恢复模块存储系统碰恢复计划。所述故障恢复计划定义一个或多个由处理器执行的函数，以恢复故障进程。处理器执行恢复计划自动恢复故障进程，无需计算机使用者参与。故障检测和恢复模块包括进程一个进程状态表，存储正常终止标识，如果进程终止正常，所述终止标识是由监控进程设置，如果由于故障导致的进程终止，则不是由进程设置。所述监控进程还包括一个事件处理器，接收来自外部监控线程的退出通知信号，用于终止进程，并转发给退出通知所述故障检测和恢复模块以决定进程是否正常终止或由于故障导致的终止。在此基础上的决定，所述故障事件检测和恢复模块可确定是否一个进程终止正常或是由于故障，请求所述

模块执行故障恢复计划。

本发明实施例还提供了一种计算机，所述计算机包括处理器;与所述处理器连接的内存。所述处理器存储一个故障检测和恢复模块确定恢复方案，由所述处理器执行的一个进程。所述故障检测和恢复模块可以自动检测系统故障的进程，并逐步恢复，从故障的进程中，执行恢复计划，而无需采取任何使用者参与的计算机行动。为进程恢复计划可以被专门定义。反过来说，恢复计划可以是一个默认计划用于恢复在没有恢复计划的进程故障。恢复计划可支配的计算机被重新启动，或进程重新启动，或者两者兼有。

在一些实施例中，监控进程储存在内存里，可进一步包括一个事件句柄进程接收退出通知信号从进程控制模块，显示该进程已终止。该事件处理器通知故障检测和恢复模块的退出通知信号，信号来自终止进程和进程控制模块。在一些实施例中，进程控制模块有一个队列存储开始和终止执行请求，由进程控制模块允许按规则执行。在一些实施例中，监控进程包括一个初始化进程配置，以确定开始和终止执行请求后，启动计算机，初始化进程输入配置的开始和终止请求进入队列为执行后，启动计算机。开始请求和终止请求在队列中可以源自不同的来源，包括另一运行进程中，连接处理器的输入装置，由一个使用者，或者是一个连接到处理器的接口单元接收请求，或者是由外部设备，或者是组合。在不同的故障进程之后，故障恢复计划可以是确定装置是否能够启动，在一个或多个进程重新启动，或两者兼而有之。故障检测和恢复模块提供的任何进程启动或重新启动初始化进程输入到执行的队列后，重新启动计算机或重新启动一个进程。其中故障恢复计划，可以是一个默认的恢复计划被用在没有故障恢复计划的进程中。监控线程被并入在一个客户端进程中，如果是由进程控制模块标识执行的进程终止，客户端进程可以包括终止进程确定在进程终止前执行进程。监控进程，或更具体地说，进程控制模块，可以包括一个为开始进程的启动进程，终止进程的终止进程。

本发明的实施例是计算机可读介质计算机进程（即监控进程），以执行一个或一个以上的上述描述的功能，监控进程由计算机执行。

附图说明

概括描述本发明内容之后，根据附图给出参考说明，所述附图无需按规定比例绘制，其中：

图 1 是本发明一个具体实施例的结构图，图中计算机含有一个存储器（即计算机可读介质），该存储器已存储了监控进程，该监控进程用于控制一个或多个进程和依照本发明内容自动从进程故障中恢复；

图 2 是本发明方法的一个具体实施例，该方法由监控进程执行，该监控进程监控和控制进程，所述进程使用进程外部的监控线程，该方法还用于执行进程故障恢复；

图 3 是本发明方法的一个具体实施例，该方法由监控进程的启动进程执行，用于启动进程和创建一个监控线程来控制进程；

图 4 是本发明方法的一个具体实施例，该方法由监控进程的终止进程执行，用于终止进程和通知退出进程终止时的故障检测结果和恢复模块；和

图 5 是本发明方法的一个具体实施例，该方法是终止进程通知用于终止的进程的监控线程，并以此来启动定制的退出进程。

具体实施方式

为使本发明的目的、技术方案和优点更加清楚，下面将结合附图对本发明作进一步地详细描述，只是对一些，并不是对所有的发明实施例进行阐述。事实上，本发明有很多表现形式，并不仅仅限于所述的表现形式。当然，阐述的这些实施例符合对应的法律要求。

定义

'和/或'是指：是指部分，或全部是由前边和后边的条件决定的。因此，"A, B, 及/或 C"，是指"A、B、C 中的一个，或者 A, B, 和 C 的全部。"

'计算机'，是用在这里，是指通常的技术属性条件。它是指一种装置具有内存和处理器接收从一个内存，使用者，或外部设备的输入数据，处理这些数据的基础上，控制进程储存在内存，处理输入数据并产生输出数据。

'计算机可读介质'是一个'内存。'

'内存'可以是一个随机存储器（RAM），只读存储器（ROM），硬盘驱动器存储单元，磁带机，光盘（光盘版），数字光盘（DVD），或者以数字或模拟形式存储的其他任何有数据储存能力的设备，无论是在光学，电子，磁力

或其他方面。'内存'包括在其定义中的任何硬件，软件或软件和硬件相结合，可以用计算机的处理器来读取，并可能存储计算机进程或数据。

'操作平台'是指一个操作系统或操作环境控制的分配和使用的硬件资源，如内存，处理器时间，硬盘空间，输入和输出单元和外围设备。操作平台包括有：windows xp， windows 2000， windows ce 系统， windows 98， windows NT， Mac 系统， Unix， Linux 系统， AJAX， I 系列， Z 系列， Solaris 和其他。

'进程'，基本上是可以由处理器执行的任何代码，。一个进程可以是一个请求进程，交互进程，或效用进程，例如。

'处理器'可以是一个微处理器，微控制器，可编程门阵列（PGA），现场可编程门阵列（FPGA），编程阵列逻辑（PAL），可编程逻辑阵列（PLA），并可能与其他形式能接收和句柄输入数据根据编程指令产生输出数据。

'发送'或'传递'是指传输数据的一个元素或进程到另一个。它可以包括传送的数据或信号，从一个元素到另一个，或有效地传输这些数据或信号，使数据句柄的一个单元或模块传输到另一单元或模块。

在一个单词后的'（s）'或'（es）'或'（ies）'，是指'一个或一个以上的意思。'因此，'process（es）'就是'一个或一个以上的进程。'

计算机 10 的概要描述

图 1 是本发明实施例计算机 10 的示意图。计算机 10 个包括一个处理器 12 个，一个内存 14，一个输入单元 16，是一个输出单元 18，可选电源 20，一个接口单元 22（IU），和通道 24。处理器 12 与内存 14 通信、输入单元 16，输出单元 18，电源 20，和接口单元 22 经过通道 24 连接进行通信。处理器 12 与外部设备、计算机 10 的接口单元 22 连接进行通信。

处理器 26

内存 14 存储一个或多个由处理器 12 执行的客户端进程 26。客户端进程或进程 26 包括一个请求进程，交互进程，效用进程，或这些进程中的多数相结合。每个客户端进程 26 由处理器 12 根据计划执行各项功能。在正常运行，处理器 12 同时运行进程 26 的时间不同。换言之，处理器的 12 根据自己的时间分配为每一个正在运行的进程分配时间和按顺序执行。

提供特殊的计算机 10 和进程 26，它可以执行，在不以任何方式有意淡化发明的最广范围，像这样的计算机和进程都可以应用，计算机 10 可实施作为一

个手持设备，例如通过使用 UPS 公司的驱动进程和包句柄的信息采集装置 (DIAD)，一个客户端进程 26 可由是信息采集装置 (DIAD) 执行，也就是请求句柄的获得和传送。这个请求可以通过编程从一个外部设备接收数据。这个数据可能显示地址，而处理器就是要提供数据传输路线，及地址，而这些数据是从手持设备中获得。这些应用可以编程允许手持设备使用输入单元 16 和输出单元 18 查看有关接收和传输的数据包，如终止接收及传输地址。它可能会进一步获得有关接收和传输的数据，如数据显示的是，包句柄设备获得或传输一个包的时间和日期，数据签名表示确认接收数据和传输数据等。此外，请求可以通过编程使驱动进程接收通过发单人发送的包单。计算机 10 以无线方式或直接通过接口单元 22 与外部装置连接，例如通过对接站，跟踪服务器等，从外部设备接收数据，并提供获得的数据传送给外部设备。获得的数据可以提供给以计费，物流，包装跟踪等为目的的外部设备。

此外，进程或进程 26，可以由一个或多个交互进程组成，使计算机 10 与外部设备通信。例如，一个交互进程，可以检测到计算机 10 是连接到对接站允许收费，其电源来源的 20 可实施作为充电电池。为响应收费站的连接事件，交互进程，可以处理器 12 执行，使计算机 10 与另一装置初始化通信，触发传输数据，或其他活动。

所述例子进程 26 存储在计算机 10 中执行。这些普通的技巧在技术中被其他请求进程，交互进程，效用进程中所应用，不管是否与接收或传输数据包有关，被计算机 10 根据请求的目的所使用。句柄的包的内容是可仿效的，本发明所揭示内容可以适用于整个与句柄包和传输包无关用途。

进程 26 提供各自的监控线索 44，其中包括进程间的通信对象的 51 (例如，一个事件，信号量，互斥，或其他机制) 与句柄 56，使进程控制模块 30 终止运行进程 26。此外，监控线程 44，可提供与终止 (TERM) 代码 60 其中确定任何句柄由进程 26 执行当监控进程 28 发出终止信号时。这种句柄可以涉及存储数据或执行其他进程终止之前的任务。对于这些特性在以下进一步详述，

监控进程 28

内存 14 存储一个监控进程 28 由处理器的 12 监控，并在必要时控制任何运行于计算机 10 的进程 26。监控进程 28 允许运行进程要求启动或终止的另一个进程。在本实施例图 1 中，监控进程 28 的有利组成是由进程 26 分开的代码。

因此，它可以控制这种过程，甚至在发生一个或一个以上的进程故障。换句话说，监控进程 28 有能力来执行计算机 11，不会受进程 26 的状况影响。在一个或多个进程 26 故障时，监控进程 28 有恢复能力。

监控进程的 28 由一个过程的控制模块 30 组成，故障检测和恢复模块 32 个，以及一个事件处理器 34。上述每个模块在下文详细讨论。

进程控制模块 30

进程控制模块 30 具有以下功能：(1)开始或终止任何的客户端进程；(2)，以确定客户是否进程运行；(3)为使另外一个客户进程去启动或终止一个客户端进程。

在一般的情况下，进程控制模块 30，包括执行开始请求 43 的开始进程 38，执行终止请求 43 的终止进程 54。进程控制模块 30 还包括一个队列 39 存放开始和终止请求 43，先入先出（先进先出）执行开始进程 38 和终止进程 54，当开始进程时为每个运行进程 26 创建一个或多个监控线程 47，使监控进程 28 监控进程故障。开始进程 38 从队列 39 接收开始请求 43，并决定根据规则 41，判断是否每个请求可以被执行。如果是的话，那将创建一个监控线程 47 外部进程 26，使这一进程受到监控，以确定它是否已经终止。如果不符合规则 41，开始进程 38，可以拒绝开始请求 43 或把它退回给排队 39 到一个或更多的突发事件发生允许进程执行。终止进程 54 句柄队列 39 中的所有终止请求 43。当从队列 39 中接到终止请求 43，用规则 70 评估请求，以决定是否可以执行终止请求。如果是的话，它标志着进程 26 中的对象 51 是目标请求，以使用 56 句柄进程间的通信对象 51，从而导致进程在终止前执行其定制终止代码 60。终止进程 54 也通知故障检测和恢复模块 32 的进程，当执行一个终止请求，以使模块获知这一进程的终止是正常的，还是由于故障导致的。反之，如果规则 70 不允许终止进程 54 执行终止请求，那么队列中的终止请求在允许执行的情况下才被执行，或者它可能完全拒绝终止请求。以何种方式终止或开始请求介绍介绍进程控制模块 30，更具体地说，队列 39，可以从详细讨论中产生几个方案。

计算机 10 个载入及执行进程控制模块 30 后启动电脑。进程控制模块 30 包括一个初始化 (INIT) 函数 36，其中定义了一系列的一个或多个进程 26，它们是由处理器 12 响应计算机 10 启动开始。此外，初始化函数 36，可确定一系列的一个或多个进程由计算机 10 重新启动后执行，这系列进程不需要重新启动

后执行同样的进程。监控进程 28 配置带有启动装置的操作平台。处理器执行监控进程 28 为定义一系列初始化进程运行初始化函数 36 发动开始进程请求。在发生不止一个进程运行时，如前所述，处理器通常多任务之间为正在运行的进程 26 分配句柄时间和正在运行的进程。

一个正在运行的进程 26 可以请求处理器 12 开始或终止另一个进程。更具体地说，客户端进程 26 可以传送请求 43 至队列 39 通过开始进程 38 启动执行，或启动终止进程 54 终止进程，一个目标进程中所指定的请求。请求 43 可以包括数据显示目标进程中为开始或终止请求和进程发起的请求。当请求 43 进到队列 39 的队尾，通过开始进程 38 或终止进程 54 分析，为适宜的类型请求的，这就决定了目标进程是否是一个由请求进程开始或终止。在此基础上由规则 41，70 所确定的一个或一个以上的几种标准，开始进程 38 确定目标进程是否可以开始或终止。在开始进程 38 和终止进程 54 的一个具体描述中，请求进程是通过规则 41，70 分配一个特权开始或终止目标进程，或者反过来说，进程可以分配一个限制，禁止它开始或终止目标进程，这取决于进程员如何分配特权和限制给进程 26。此外，如果有两个或两个以上被确定是目标进程，它可能是开始进程 38 的规则 41 或终止进程 54 的规则 70 可以编程评估相对优先的起始进程的请求。例如，如果一个开始请求由一个进程发起，处理器 12 可以执行监控进程 28，以确定是否按规定 41，一个进程先前发出的终止请求 43 当收到开始请求时仍然有效成为目标进程，此进程具有从提出请求的进程中有优先权的请求。如果是的话，那么处理器 12 可以拒绝开始请求进程开始请求 43，或可暂缓执行，直到发生一个或一个以上的事件，如终止进程终止请求具有更优先的请求。然后，再由处理器 12 执行开始进程 34 以导致请求进程开始。此外，监控进程 28 利用内部队列 39 使开始请求和终止请求按顺序执行。如有需要，消除不相干的请求，使队列优化，例如，如果一个开始进程的请求在一个终止请求后，开始就请求可能会终止以节约处理时间。

此外，当通过输入单元 16 从一个使用者收到请求 43，进程 26 开始或终止。例如，一个使用者的计算机 10 可以操作输入装置 16 产生的请求 43 提供给处理器 12 开始或终止一个或多个进程 26。队列 39 拥有请求 43 直到根据规则 41，70 由开始进程 38 或终止进程 54 执行。

然而，另一种方式，进程 26 可以开始或终止响应外部设备产生的请求 43，

请求 43 由处理器 12 通过接口单元 22 接收。请求 43 在队列 39 中直到由开始进程 38 根据规则 41 或终止进程 54 根据规则 71 执行。

在图 1 中所述类型的请求 43，作为外部启动/终止请求 55，因为它们源自或引发外界的进程控制模块 30。

除上述初始化函数 36 和外部启动/终止请求 43，开始或终止请求 43 可来自故障检测和恢复模块 32。以何种方式出现，在稍后部分对故障检测和恢复模块 32 的详细细节进行论述。

当进程 26 被启动，操作平台 46 创建一个句柄 48 确定该进程，并提供同样的开始进程 38。开始进程 38 监控句柄 48，由一个监控线程 47 创建和存储在进程控制模块 30 中。当一个进程 26 终止，操作平台 46 无效其句柄 48，造成监控线程 47 警报进程已经终止。这会导致监控线程 47 产生退出通知信号，给事件处理器 34 由故障检测和恢复模块 32 着手检查，以确定进程的终止是正常还是故障导致终止。

如前所述，一个客户端进程 26 可以修改，就如它可以由监控进程 28 控制。终止请求从监控进程 28 至客户端进程 26 的通信是由唯一的系统标识符支持，它可以由一个事件，信号量，互斥，或其他机制实施计算机中进程间的通信。一个监控线程（"ST"）44，被编程到客户端进程 26 -R，在等待由监控进程 28 发信号给对象 51（例如，一个事件，信号量，互斥，或其他机制）。当监控进程 28 执行一个终止请求，它利用句柄 56 发信号给对象 51，启动监控线程 44，以激活服务终止（TERM）函数 60，这个函数可被编程到终止进程 26。当客户端进程被终止，无效进程句柄 48 由操作平台 46 发出信号给对应的监控线程 47，在监控进程 28 中生成退出通知传送到事件处理器 34，从而激活故障检测和恢复模块 32。

在正常操作下，有很多运行于计算机 10 的客户端进程 26。因此，在图 1 中监控线程 47-1 通过 47-x 对应于每一个过程 26-R1，通过 26-RX 运行于计算机 10。'R'表示进程 26 正在运行和'X'代表运行于计算机 11 的客户端进程的数量。监控线程 47-1 通过 47-X 继续运行在进程控制模块 30，只要他们对应的进程 26 正在运行。在图 1 中，这些没有运行的进程由数字'26-NR1-26-NRQ'标识，'NR'标识进程的状态是没有执行，'Q'是没有被执行的进程编号。

当执行终止请求 43，终止进程 54 设置一个正常终止标识对应故障检测和

恢复模块 32 中的进程状态表 77 中的进程 26。设置这个标识，表明进程 26 已由终止进程请求终止。当一个进程 26 终止运行，操作平台 46 使进程句柄 48 无效。将这种句柄无效发信号给对应的进程控制模块 30 中的监控线程 47，使之产生退出通知信号给事件处理器 34 从而警报故障检测和恢复模块 32 检查进程是否正常终止还是故障终止，正常终止标识存储在进程中状态表 77。故障检测和恢复模块 32 和操作的细节在下面阐述。

故障检测和恢复模块 32

故障检测和恢复模块 32 提供下列功能：（1）它允许一个客户端进程配置如何使模块恢复进程故障事件；（2）模块能够保持所有运行在计算机 10 的使用者进程的状态（即，数据显示进程是否在运行，是否正常终止或故障终止）；（3）模块监控客户端进程的故障；（4）在检测故障进程后，模块为故障进程执行配置的恢复计划。

如图 1 所示，故障检测和恢复模块 32 存储故障恢复计划 75-1 通过 75 -Y（Y 是计划的数目），为计算机 10 中的一个或多个进程 26。故障恢复计划 75-1 至 75 -Y 可专门配置一个或一个以上对应的进程 26，允许计算机 10 以一个定制的方式从故障进程 26 中恢复。举例来说，在故障进程后，故障恢复计划 75 可以指定计算机 10 恢复故障进程 26 是通过重新启动计算机或者重新启动故障进程，或者两者兼有。如果一个定制的计划 75 是没有界定为一个进程 26，那么故障检测和恢复模块 32 使用默认计划 75 -Z 恢复故障进程。

故障检测和恢复模块 32 拥有进程状态表 77 显示哪些进程正在运行，而哪些进程没有正在运行。故障检测和恢复模块 32 拥有正常终止数据，正常终止数据来自终止进程 54，终止进程 54 为任何进程正常终止以反映终止是不是故障导致的。如果发现有任何进程终止，是否正常终止或由于故障，操作平台 46 发信号给监控进程的监控线程 47，为这个进程，通过无效进程句柄 48，然后通知事件处理器 30 途经退出通知事件激活故障检测和恢复模块 32。为了响应退出通知信号，故障检测和恢复模块 32 更新进程状态表 77，表示进程生成退出通知已经终止。此外，故障检测和恢复模块 32 在进程状态表 77 中检查对应的正常终止标识，以确定该进程是否已正常终止。如果是的话，进程检测和恢复模块 32 需要执行，没有进一步处理的终止进程。反过来说，如果进程终止标识显示进程终止异常，然后故障检测和恢复模块 32 执行故障恢复计划 75 对应的故障

进程以恢复故障。如前所述，故障恢复计划 75 可专门配置一个进程 26 以确定如何在计算机恢复故障进程。在一个对应的故障进程完全恢复后，故障恢复计划 75 是图 1 从 75-1 至 75-Y 中指定一个，指明计算机 10 是否应启动，故障进程重新启动，或两者兼而有之。此外，这种计划 75 可以被定义表明是否有其他进程必须重新开始或重新启动，以恢复故障。故障恢复计划 75 并不是专门为进程 26 定义的，故障检测和恢复模块 32 用默认恢复计划 75-Z 恢复运行在计算机 10 上的故障和进程。为故障进程执行故障恢复计划 75 导致产生一个或更多的开始或终止请求 43，检测和恢复模块 32 规定更新初始化函数 36，队列 39，或者两者兼有。

事件句柄 34

事件处理器 34 响应事件所产生在监控进程 28 内部的监控线程 47。更具体地说，当一个进程的 26 通过执行终止函数 60 和终止代码 61 退出，终止进程发信号给监控进程 28 的监控线程 47 由句柄 48 无效终止进程，从而导致监控进程产生退出通知事件通知事件处理器 34，终止进程 26。此外，如果一个进程 26 由于故障终止异常，监控进程 28 以类似的方式通知事件处理器 34。响应退出通知，事件处理器 34 激活故障检测和恢复模块 32 确定进程是否正常终止或是故障终止，在进程 26 退出前，进程状态表 77 中的正常终止标识或不是正常终止标识由终止进程 54 设置。

使用监控线程监控和控制进程的方法

图 2 是使用外部运行在计算机 10 上的进程相关联的监控线程 47 监控一个或多个进程 26 的流程图。监控进程 28 被编程由图 1 中的计算机 10 执行，以实现图 2 中的方法步骤。

图 2 中的步骤 200，装置 10 启动，步骤 202，为响应启动，装置 10 执行一个初始化（初始化）进程 36。这会导致处理器 12 启动由初始化进程 36 定义的一个或多个进程 26。更具体地说，通过其执行，初始化进程 36 发送一个或多个开始请求 23 为开始进程 38 的目标进程 26。初始化进程 36 编程启动特殊的进程 26。步骤 204，开始进程 38 从队列 39 收到启动请求（s）43 并同时执行。当每个进程 26 开始，为启动进程 26，开始进程 38 从装置的操作平台 46 收到句柄 48。每个句柄 48 有能力确定各自的外部进程 26-R 的运行状态。客户端进程 26 编程创建一个监控线程 44 开始并把这种监控线程 44 连同进程间通信对象

51 和句柄 56 合并到进程 26 中。另外，开始进程 38 生成监控线程 47，在内部合并对象 53 与句柄 48，因此，使监控进程 28 可以监控进程 26 在外部的运行状态。步骤 206，开始进程 38 收到由各种可能的来源所产生的外部开始请求 (s) 43，包括另一个运行进程 26，使用者的操作输入单元 16，一个通过接口单元 (IU) 22 的外部设备，或结合所述。开始进程 38 可以执行时在先处理队列请求中的开始请求 (s) 43。假设请求 (s) 43 可按照规则 41 执行，开始进程 38 启动进程 (es) 26，从操作平台 46 获得对应的句柄 (s) 48，并创建一个监控线程 (s) 47，合并各自的对象 53 和句柄 48，使进程 (es) 26 可以被监控。如上所述，进程 26 创建它们自己的监控线程 44，合并从操作平台 46 接收的进程间的通信对象句柄 56，使进程控制模块 30 控制进程 26。

步骤 208，事件处理器 34 等待来自监控线程 (s) 47 的退出通知，响应对应通过操作平台 46 无效的句柄 (s) 48，显示对应进程的 26 已经终止。步骤 210，为响应事件处理器的退出通知检测，故障检测和恢复模块 32 执行，以确定是否退出通知是由于正常终止还是进程 26 的故障终止。模块 32 执行检测函数以确定终止进程 54 是否在进程 26 终止之前在进程状态表 77 中设置了一个正常的终止标识。如果终止进程 54 设置了正常终止标识，进程没有故障返回控制给步骤 208 其中处理器 34 继续监控每个运行进程退出通知。反过来说，如果步骤 212 进程没有在进程状态表 77 中设置正常终止标识，那么这个进程是由于故障终止的，此情况下，仍然控制与故障检测和恢复模块 32 更新一套初始化进程 26，以重新启动基于存放在进程状态表 77 中重新启动标识的值。它还根据故障进程 26 的故障恢复计划 75 设置启动和/或重新启动标识。如果没有为故障进程 26 指定恢复计划 75，那么模块 32 用默认计划 75-Z 设置 (或重新设置) 启动和/或重新启动标识以恢复故障。步骤 216，模块 32 确定是否故障恢复计划请求重新启动计算机 10。如果是的话，由处理器 12 执行步骤 200 重新启动装置。反过来说，如果没有设置重新启动标识，则进入步骤 218，处理器 12 执行模块 32 引起重新启动进程，标识需要为计划 75 被执行重新启动进程状态表 77。如果处理器的执行模块 32 表明没有进程需要重新启动，则进入步骤 208，事件处理器 34 继续监控运行进程退出通知。如果模块 32 决定步骤 218 重新启动是必要的，在步骤 204 则模块 32 传递开始请求 43 排队等待开始进程 38 执行以启动步骤 214 中

恢复计划指定的进程。执行步骤 204 后，控制进入步骤 208，事件处理器 34 使用对应的监控线程 47 监控正在运行的进程 26。

图 3 是本发明方法的一个具体实施例，所述方法用于启动装置 10 中的进程 26。步骤 300，监控进程或客户端进程 26-R 发起开始请求 43 请求启动目标进程 26-NR。开始请求 43 存储在队列 39 中等待开始进程 38 执行。步骤 302，开始进程 38 启动新的进程 26。开始进程 38 请求操作平台 46 启动请求的进程 26。开始进程 38 从操作平台 46 接收开始进程的句柄 48，并将句柄保存在内存。客户端进程 26 合并一个监控线程 44、一个对象 51 和一个句柄 56，使它能够被监控进程控制，步骤 304，开始进程 38 在监控进程 28 内创建了一个静态的监控线程 47、一个对象 53 和一个句柄 48，当开始客户端进程的句柄被无效时，静态的监控线程 47、对象 53 和句柄 48 被激活（即，当进程终止）。激活的监控线程 47 使它发出退出通知给事件处理器 34，事件处理器 34 顺序警告故障检测和恢复模块 32 检查是否进程的退出是正常的，或因故障导致退出。

图 4 是本发明方法的一个具体实施例，所述方法用于终止装置 10 中的进程 26。步骤 400 监控进程 28 或客户端 26-R 发出终止请求 43 给另一个进程 26-NR。终止请求 43 排队等候，直至它可以由终止进程 54 执行。步骤 402 终止进程 54 确定进程被终止后是否应该重新启动。它做这个决定是根据规则 70，故障恢复计划 75，或者两者兼有。步骤 402 如果终止进程 54 确定进程不应被重新启动，那么进入步骤 404，它给进程设置了对应的标识以表示进程在终止后没有被重新启动。相反，在步骤 402 中，如果终止进程确定进程被终止后应该重新启动，进入步骤 406，则在表 77 中设置对应的标识表示这样的进程在被终止后应该被重新启动。步骤 408，终止进程 38 由处理器 12 执行确定装置 10 在目标进程终止后是否应该重新启动。如果不是的话，进入步骤 410，终止进程 54 设置对应的标识表示装置 10 在目标进程终止后不应该被重新启动。反过来说，在步骤 408 中如果终止进程 54 确定装置 10 在目标进程终止后应该重新启动，则进入步骤 412，终止进程 54 设置对应的标识在进程状态表 77 中。步骤 414，终止进程 54 发信号给客户端进程的监控线程 44 中的进程间通信对象 51，同样发出终止请求到目标进程 26。

图 5 是本发明方法的一个具体实施例，所述方法用于终止在装置 10 中执行的客户端进程 26。步骤 500，客户端进程 26 根据自己的代码正常执行客户端

句柄。没有终止请求，步骤 500 中正常客户端句柄通过正常终止过程 26。然而，在这种情况下，步骤 502，终止进程 54 为进程 26 发送一个终止请求给监控线程 44。终止请求通过发信号给监控线程 44 的进程间通信对象激活进程 26 的监控线程 44。步骤 504，在执行的终止代码 61 之前终止执行之前，根据编程 60 进程 26 定制退出处理。

可选择的方式

在不脱离本发明原理的前提下，还可以做出若干改进和润饰，这些改进和润饰也视为本发明的保护范围。举例来说，故障恢复计划 75 并不一定要存储在故障检测和恢复模块 32 中，这些计划可以存储在内存 14 的任何地方，除在揭示的模块之外或在各自进程 26 的监控线程 44 或 47 内。同样地，当一个进程 26 通过对应的监控线程 44 发信号退出时，终止代码 60 定义代码执行，监控线程 44 与正在运行的进程 26 分开存储，当监控线程 44 被发信号退出时进程 26 可被重新获得执行。进程状态表 77 不必被存储在故障检测和恢复模块 32，但可被存储在内存 14 的另一个容易到达模块 32 和事件处理器 34 位置。虽然规则 41，70 通常是与进程 26 分开存储的，本规则适用于某一特定进程 26 可被存储在这样的进程内，或者存储在有可能不利遮蔽规则的相关监控线程 44。如果没有偏离范围的发明，进程控制模块 30，事件处理器 34 和故障检测和恢复模块 32 是可以相互结合成较大的或划分成较小的代码段根据他们的函数作用。这些和其他的修改在发明的范围内被理解。

本发明在这里阐述的修改和具体实施例在技术上被理解，在上述描述和相关附图提出了一些有益效果。因此，以上所揭露的仅为本发明一种较佳实施例而已，当然不能以此来限定本发明之权利范围，因此依本发明权利要求所作的等同变化，仍属本发明所涵盖的范围。

产业适用性

本发明各实施例所揭露和要求保护的计算机及介质，通过使计算机自动检测和恢复进程的故障，而具有产业适用性。

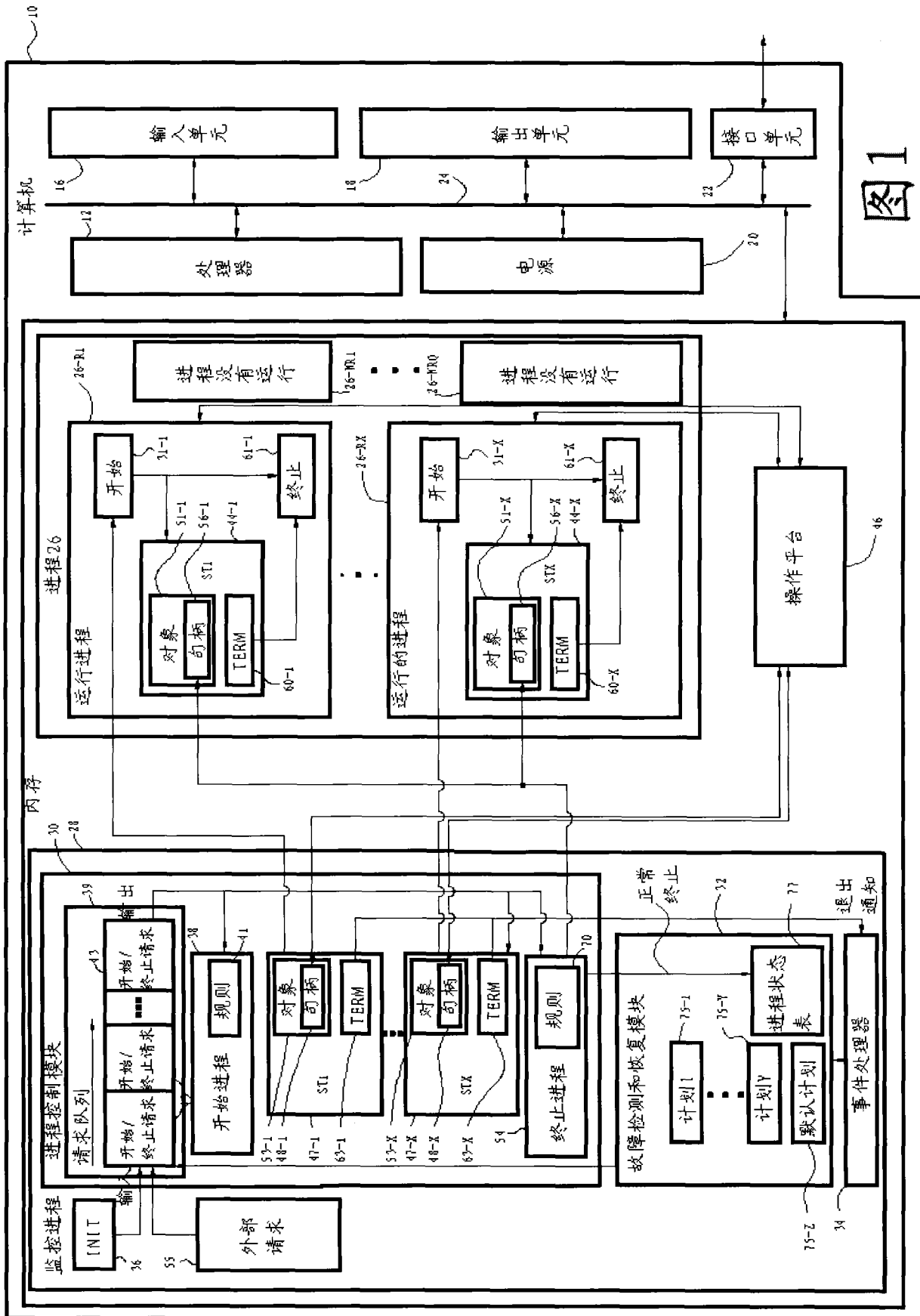


图1

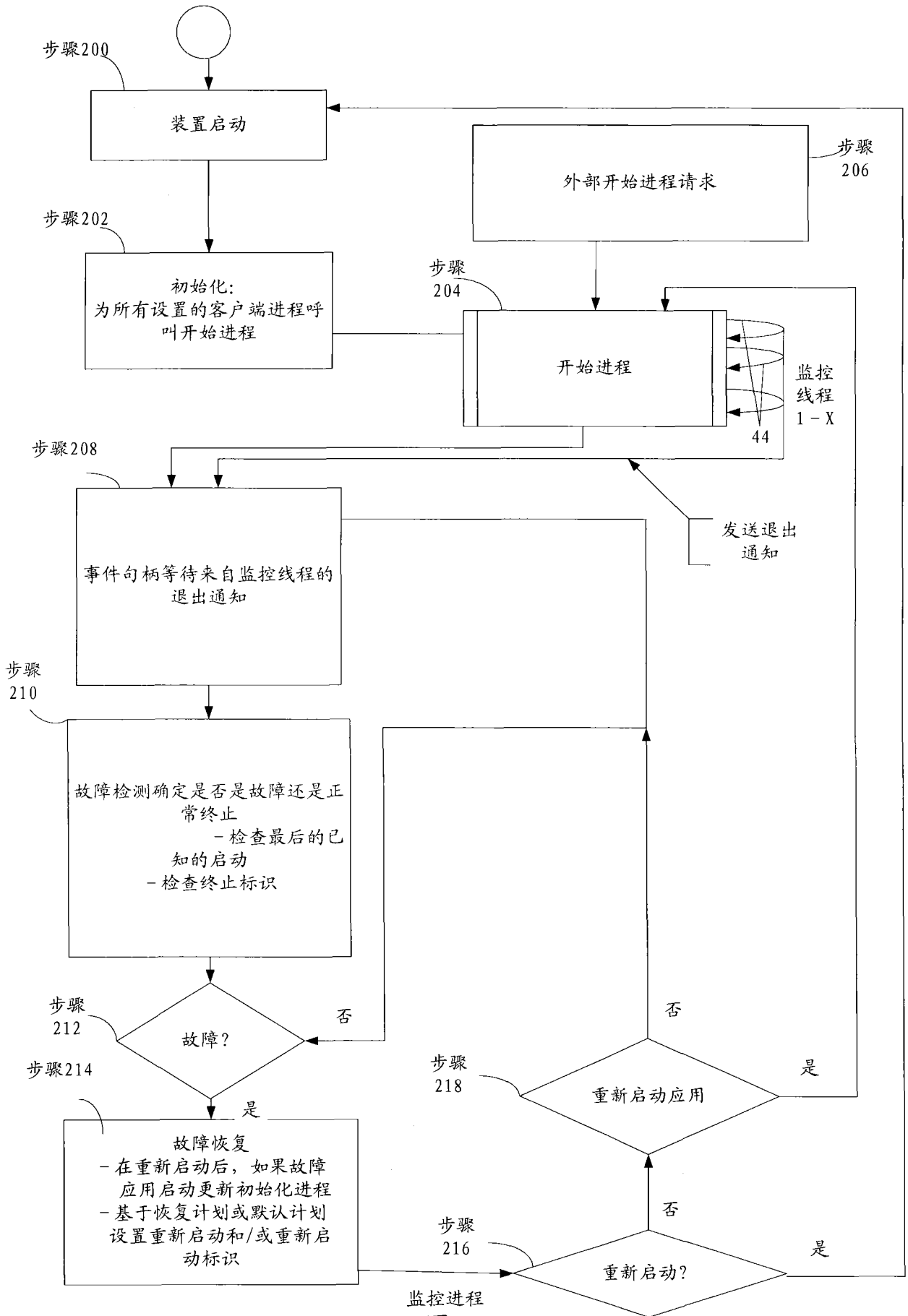


图2

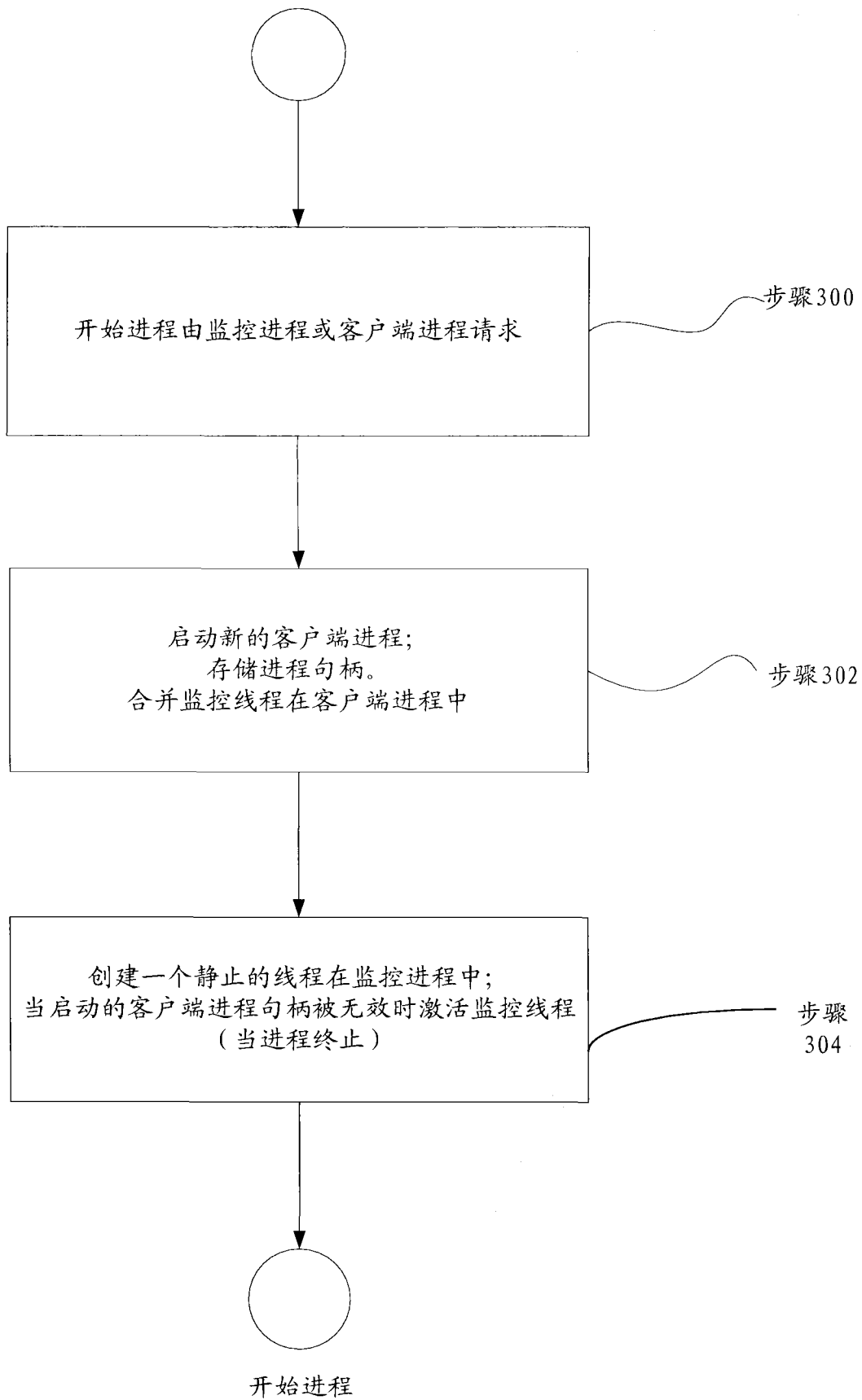
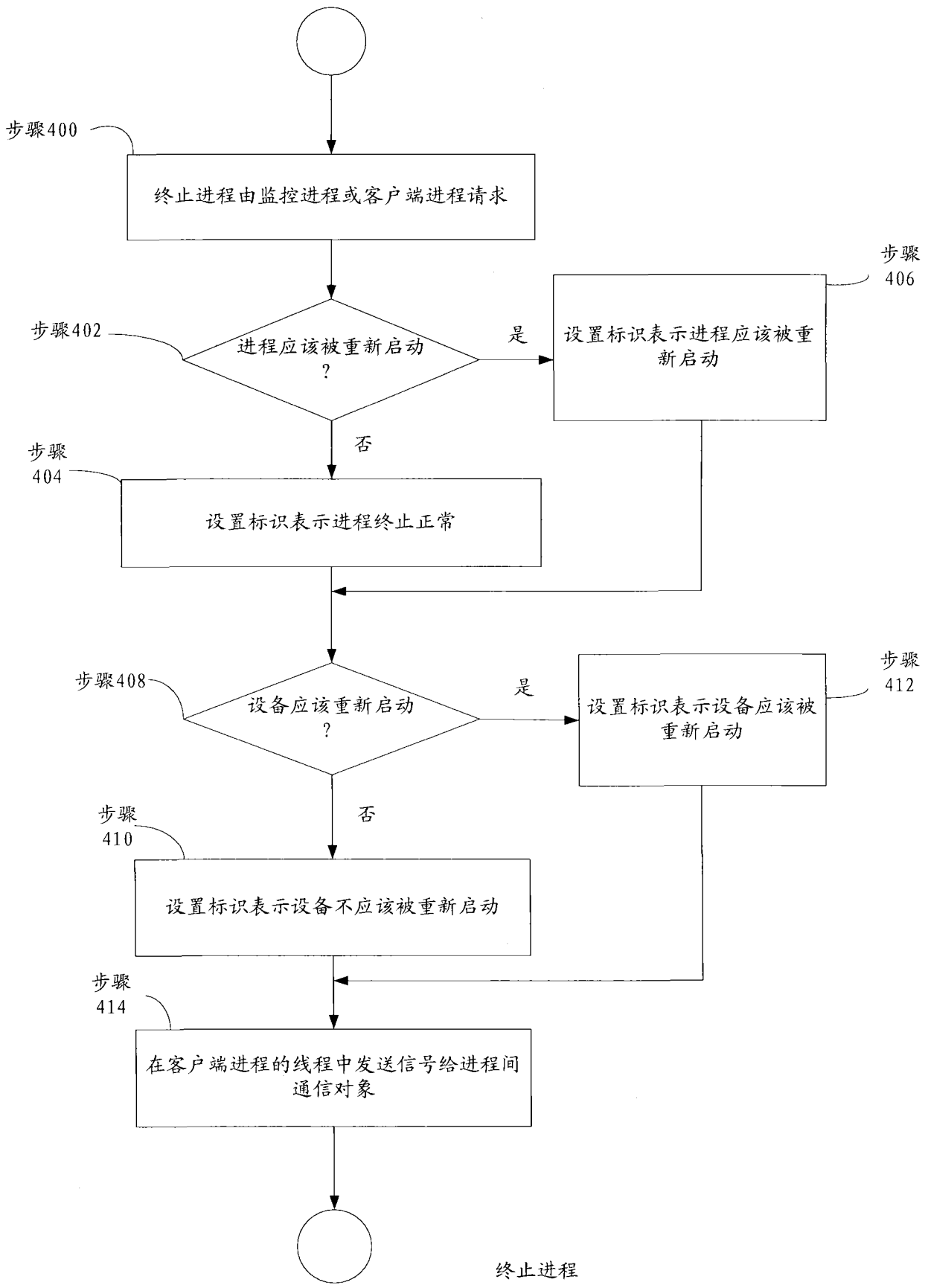


图 3



终止进程

图4

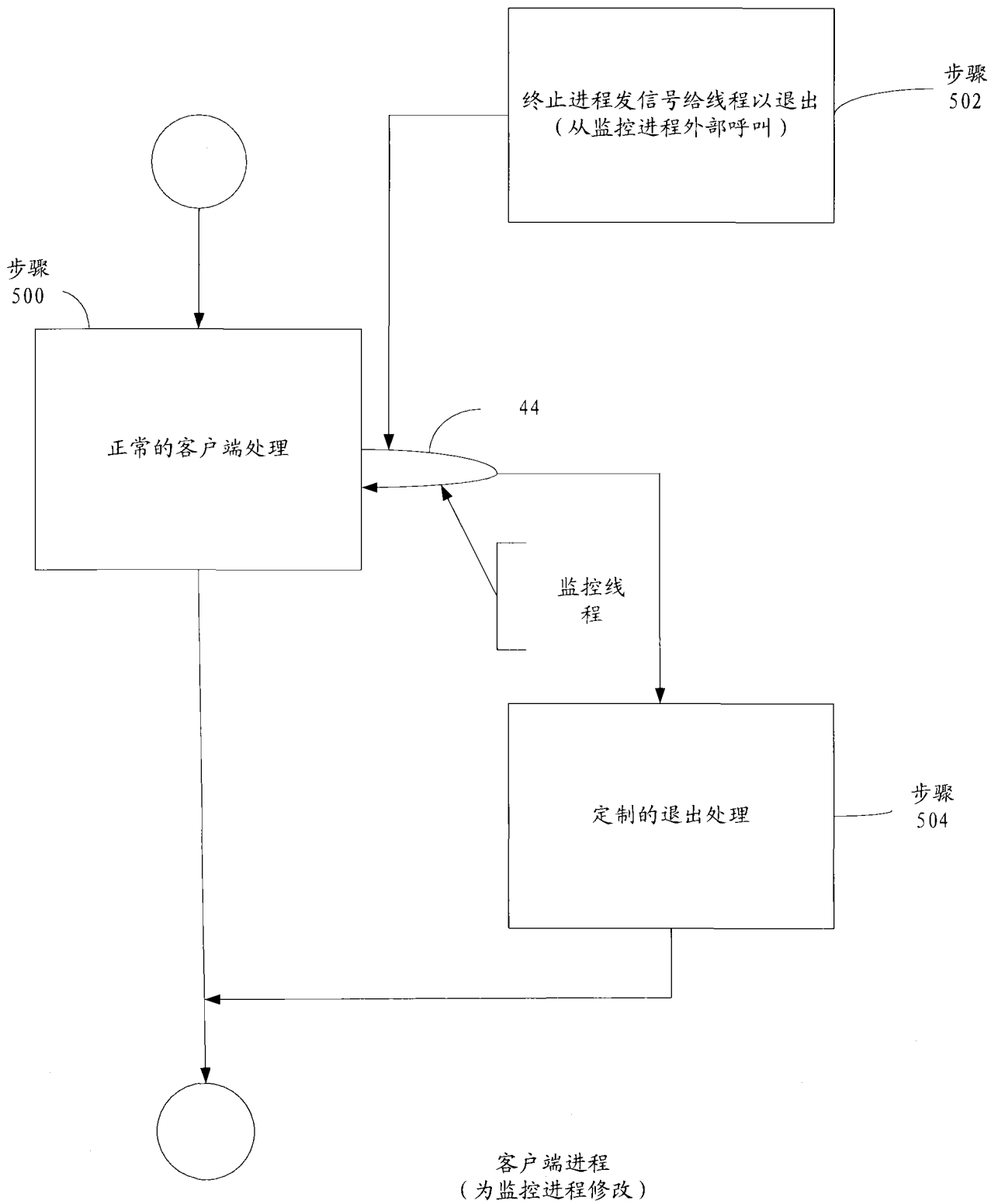


图5