



(12)发明专利

(10)授权公告号 CN 105988866 B

(45)授权公告日 2019.08.23

(21)申请号 201610147017.9

(51)Int.Cl.

(22)申请日 2016.03.15

G06F 9/46(2006.01)

(65)同一申请的已公布的文献号

审查员 罗捷

申请公布号 CN 105988866 A

(43)申请公布日 2016.10.05

(30)优先权数据

2015-053658 2015.03.17 JP

(73)专利权人 佳能株式会社

地址 日本东京都大田区下丸子3-30-2

(72)发明人 母里南阳

(74)专利代理机构 北京怡丰知识产权代理有限公司

公司 11293

代理人 迟军

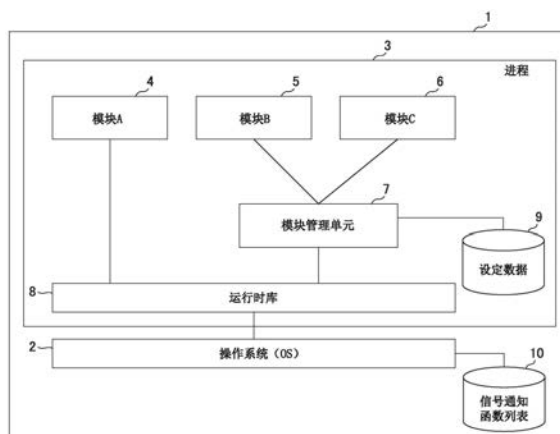
权利要求书3页 说明书9页 附图10页

(54)发明名称

信息处理装置及其控制方法

(57)摘要

本发明提供一种信息处理装置及其控制方法。该信息处理装置能够执行由多个模块组成的程序,所述多个模块包括受模块管理单元的管理的自模块和不受管理的他模块。所述模块管理单元(7)进行在程序中发生例外时信号要被通知到的函数的登记。当在例外发生的情况下接收到信号时,执行例外处理。当加载进程中的特定模块时,如果他模块的函数已被登记,则所述模块管理单元存储该函数的信息,并取而代之登记用于执行自模块的函数的例外处理的函数。针对存储的他模块的函数,发送与已登记的函数接收到的信号相同的信号。



1. 一种信息处理装置,所述信息处理装置包括:

登记单元,其用于将模块的函数登记为信号通知函数,该信号通知函数在程序中发生例外的情况下接收信号;以及

确定单元,其用于确定所述函数是否已被登记为所述信号通知函数;

其中,在所述登记单元将第一模块的第一函数登记为信号通知函数的情况下,所述确定单元确定第二模块的第二函数是否已被登记为信号通知函数,并且在所述确定单元确定所述第二函数已被登记为信号通知函数的情况下,所述登记单元将用于调用所述第一函数和所述第二函数的第三函数,登记为信号通知函数,

其中,所述第一模块是受所述登记单元的管理的模块,并且所述第二模块是不受所述登记单元的管理的模块。

2. 根据权利要求1所述的信息处理装置,

其中,所述登记单元通过区分所述第一模块与所述第二模块,来进行模块的管理,并且

其中,在所述确定单元确定所述第二函数被登记为信号通知函数的情况下,所述登记单元将所述第二函数的信息存储在存储单元中。

3. 根据权利要求2所述的信息处理装置,所述信息处理装置还包括:

执行单元,其用于当所述例外发生时,通过向登记的函数发送所述信号来执行例外处理,

其中,当所述执行单元通过向所述登记的函数发送所述信号来执行例外处理时,所述登记单元将用于调用所述第一函数和所述第二函数的第三函数,发送给存储在所述存储单元中的所述第二函数。

4. 根据权利要求1所述的信息处理装置,其中,在所述登记单元将所述第一函数登记为信号通知函数的情况下,所述确定单元确定所述第一函数是否已被登记为信号通知函数,并且在所述确定单元确定所述第一函数尚未被登记为信号通知函数的情况下,所述登记单元将所述第三函数,登记为信号通知函数。

5. 根据权利要求1所述的信息处理装置,其中,所述登记单元进行将所述第一函数的信息存储在第一信息管理区域中并将所述第二函数的信息存储在第二信息管理区域中的处理。

6. 根据权利要求1所述的信息处理装置,其中,在所述登记单元将所述第一函数登记为信号通知函数的情况下,所述确定单元确定所述第二函数是否已被登记为信号通知函数,并且在所述确定单元确定所述第二函数已被登记为信号通知函数的情况下,所述确定单元进一步确定所述第一函数是否已被登记为信号通知函数,并且在所述确定单元确定所述第一函数尚未被登记为信号通知函数的情况下,所述登记单元进行将已登记的所述第二函数的信息存储在第二信息管理区域中的处理。

7. 根据权利要求2所述的信息处理装置,所述信息处理装置包括:

多个登记单元,

其中,所述多个登记单元共享在例外处理时调用的所述函数的设定信息,如果不存在所述设定信息,则创建新的设定信息,而如果存在所述设定信息,则将所述登记单元的条目信息登记在所述设定信息中。

8. 根据权利要求7所述的信息处理装置,其中,如果在所述设定信息中存在第二登记单

元的条目信息,则所述多个登记单元当中的第一登记单元不删除所述设定信息,而删除所述设定信息中的所述第一登记单元的条目信息。

9.根据权利要求2所述的信息处理装置,其中,所述登记单元获取所述信息处理装置的资源信息,根据所述资源信息控制当所述例外发生时要调用的函数的顺序,并且将当所述例外发生时要调用的函数的顺序改变成所述模块的优先级的顺序。

10.一种由信息处理装置执行的控制方法,所述方法包括:

登记步骤,将模块的函数登记为信号通知函数,该信号通知函数在程序中发生例外的情况下接收信号;以及

确定步骤,确定所述函数是否已被登记为所述信号通知函数;

其中,在所述登记步骤中将第一模块的第一函数登记为信号通知函数的情况下,在所述确定步骤中确定第二模块的第二函数是否已被登记为信号通知函数,并且在所述确定步骤中确定所述第二函数已被登记为信号通知函数的情况下,将用于调用所述第一函数和所述第二函数的第三函数,登记为信号通知函数,

其中,所述第一模块是受所述登记步骤的管理的模块,并且所述第二模块是不受所述登记步骤的管理的模块。

11.根据权利要求10所述的由信息处理装置执行的控制方法,

其中,在所述登记步骤中,区分并管理所述第一模块和所述第二模块,并且

其中,在所述确定步骤中确定所述第二函数已被登记为信号通知函数的情况下,将所述第二函数的信息存储在存储单元中。

12.根据权利要求11所述的由信息处理装置执行的控制方法,该方法还包括:

执行步骤,当所述例外发生时,通过向登记的函数发送所述信号来执行例外处理,

其中,当在所述执行步骤中通过向所述登记的函数发送所述信号来执行例外处理时,在所述登记步骤中,将用于调用所述第一函数和所述第二函数的函数,发送给存储在所述存储单元中的所述第二函数。

13.根据权利要求10所述的由信息处理装置执行的控制方法,其中,在所述登记步骤中将所述第一函数登记为信号通知函数的情况下,在所述确定步骤中确定所述第一函数是否已被登记为信号通知函数,并且在所述确定步骤中确定所述第一函数尚未被登记为信号通知函数的情况下,将所述第三函数,登记为信号通知函数。

14.根据权利要求10所述的由信息处理装置执行的控制方法,其中,在所述登记步骤中,进行将所述第一函数的信息存储在第一信息管理区域中并将所述第二函数的信息存储在第二信息管理区域中的处理。

15.根据权利要求10所述的由信息处理装置执行的控制方法,其中,在所述登记步骤中将所述第一函数登记为信号通知函数的情况下,在所述确定步骤中确定所述第二函数是否已被登记为信号通知函数,并且在所述确定步骤中确定所述第二函数已被登记为信号通知函数的情况下,在所述确定步骤中确定所述第一函数是否已被登记为信号通知函数,并且在所述确定步骤中确定所述第一函数尚未被登记为信号通知函数的情况下,进行将已登记的所述第二函数的信息存储在第二信息管理区域中的处理。

16.根据权利要求11所述的由信息处理装置执行的控制方法,其中,在存在多个登记单元的情况下,在所述登记步骤中,共享在例外处理时调用的所述函数的设定信息,如果不存

在所述设定信息,则创建新的设定信息,而如果存在所述设定信息,则将所述登记单元的条目信息登记在所述设定信息中。

17.根据权利要求16所述的由信息处理装置执行的控制方法,其中,在所述登记步骤中,如果在所述设定信息中存在第二登记单元的条目信息,则所述多个登记单元当中的第一登记单元不删除所述设定信息,而删除所述设定信息中的所述第一登记单元的条目信息。

18.根据权利要求11所述的由信息处理装置执行的控制方法,其中,在所述登记步骤中,获取所述信息处理装置的资源信息,根据所述资源信息控制当所述例外发生时要调用的函数的顺序,并且将当所述例外发生时要调用的函数的顺序改变成所述模块的优先级的顺序。

信息处理装置及其控制方法

技术领域

[0001] 本发明涉及用于在信息处理装置的进程终止时的例外处理的技术。

背景技术

[0002] 在由信息处理装置执行的程序中,存在由于输入偏离预定范围的值、存储器的访问冲突等而发生未预料到的例外的情况。例如,存在如下的情况,在程序的执行期间进程最终异常终止时,存储器中保持的数据丢失。在操作期间未被保存的数据(例如登录信息或注册表信息等)将消失。作为其对策,使用由操作系统(在下文中,也将被称为OS)提供的机制,能够在进程异常终止之前检测例外,并且能够获取进程的转储信息。这是程序员调查例外的原因并且修改程序的有效措施。

[0003] 作为在现有技术中针对进程的异常终止而使用的例外处理方法,存在如下的方法,即当加载特定模块时,将定义了当例外发生时期望执行的处理的函数登记在OS中。当进程检测到异常时,OS向登记的函数发送信号。例如,通过将定义了转储信息获取处理的函数登记在OS中,能够获取转储信息。作为在当例外处理变得必要时将接收信号且因而要执行例外处理的情况下进行的登记方法,存在日本特开平7-334377号公报中公开的方法。该方法使得信号设定所需的程序员的负担能够得到缓解,并且使得例外处理能够可靠地且容易地如预期进行。当控制从一个模块转移到另一个模块时,进行如下的处理,即转存(evacuate)例外处理的登记内容,并且当控制返回时恢复已被转存的信息。

[0004] 在日本特开平7-334377号公报中公开的技术中,没有提及关于当接收到信号并且执行例外处理时的函数的调用(calling)控制。例如,假设当程序包括由除其创建者之外的人开发的模块(在下文中,也将被称为他模块)的情况下,进行例外处理。一般的程序由多个模块组成。随着程序的规模变得越来越大,越来越多数的人或越来越多的组参与其中,因此,很少有单个程序员或团队实现了所有模块的情况。对于他模块,由于仅公开了其界面的一部分,但其其余部分未公开,因此存在与无法被控制的模块相结合而构成了程序的情况。由于这个原因,当在例外处理时,首先调用不在控制下的模块的函数时,依据该模块的控制,可能存在信号不被发送到由程序的创建者开发的模块的情况。在这种情况下,存在仅能获取某些模块的转储信息的可能性。

发明内容

[0005] 本发明提供一种能够执行当例外发生时正执行的进程中加载的模块的例外处理的信息处理装置及其控制方法,以及记录介质。

[0006] 根据本发明的装置具有:登记单元,其用于将模块的函数登记为信号通知函数,该信号通知函数当在程序中发生例外的情况下接收信号,以及确定单元,其用于确定是否已有函数被登记为所述信号通知函数。在所述登记单元将第一模块的函数登记为所述信号通知函数的情况下,所述确定单元确定第二模块的函数是否已被登记为所述信号通知函数,并且在所述确定单元确定所述第二模块的函数已被登记为所述信号通知函数的情况下,所

述登记单元将用于调用所述第一模块的函数和所述第二模块的函数的函数,登记为所述信号通知函数。

[0007] 通过以下参照附图对示例性实施例的描述,本发明的其他特征将变得清楚。

附图说明

[0008] 图1是示出本发明的实施例中的信息处理装置的结构示例的框图。

[0009] 图2是示出信息处理装置的硬件结构的示例的框图。

[0010] 图3是示出模块管理单元管理的设定数据的结构示例的图。

[0011] 图4是示出OS管理的信号通知函数列表的结构示例的图。

[0012] 图5是示出第一实施例中当加载模块时由模块管理单元进行的处理的示例的流程图。

[0013] 图6A和图6B是示出第一实施例中当加载模块时的数据转变的示例的图。

[0014] 图7是示出第一实施例中当例外发生时由模块管理单元进行的例外处理的示例的流程图。

[0015] 图8是示出第二实施例的信息处理装置的结构示例的框图。

[0016] 图9是示出第二实施例中由模块管理单元进行的函数信息登记处理的示例的流程图。

[0017] 图10是示出第二实施例中由模块管理单元进行的设定数据删除处理的示例的流程图。

[0018] 图11是示出第三实施例中当例外发生时由模块管理单元进行的处理的示例的流程图。

具体实施方式

[0019] 下面,将使用附图描述本发明的各个实施例。在参照图1至图4描述在各个实施例中是共同的事项之后,将描述各个实施例的详情。

[0020] 图1是示出本发明的实施例中的信息处理装置1的系统结构示例的框图。信息处理装置1是用于执行各种类型的程序的执行装置,并且多个进程3在信息处理装置1的操作系统(OS)2上运行。OS 2是管理硬件和资源的基本软件组,并且OS 2的典型示例包括Microsoft Windows、Linux等。各个进程3是执行程序的单位,并且被假设为由例如多个模块4至6、模块管理单元7以及运行时库(runtime library)8组成。图1中示出的模块4至6表示模块A至C。各个模块表示比进程3更小的执行程序的单位。

[0021] 在本实施例中,存在两种类型的模块,即“自模块(self-module)”和“他模块(other module)”。“自模块”是表示模块管理单元7能够管理的模块的第一类型模块。而“他模块”是表示在模块管理单元7的管理以外的模块的第二类型模块。“他模块”包括由除程序的创建者之外的人开发的模块。如图1中例示,模块A是他模块,与运行时库8进行直接的信息的发送和接收。而模块B和模块C是自模块,与模块管理单元7进行信息的发送和接收。

[0022] 模块管理单元7具有管理自模块的功能。在此提及的管理主要是指将各个模块的例外处理时要调用的函数登记在OS 2中,或者当在进程3中检测到异常时,执行各个模块的例外处理。此外,模块管理单元7管理设定数据9。下面,将使用图3描述作为模块管理单元7

管理的设定信息的设定数据9的详情。

[0023] 运行时库8是由多个程序共享的功能被编译的模块,并且是在执行程序时的必要库。当执行开发的程序时,运行时库8被结合地使用,并且运行时库8与OS 2交换信息。

[0024] 图2是示出信息处理装置1的硬件结构的示例的框图。中央处理单元(CPU) 21读取、解释并执行预定的程序。CPU进行如下的处理,即执行存储在只读存储器(ROM) 26内部的程序区域中的程序、从硬盘23加载到随机存取存储器(RAM) 22的OS 2、以及诸如通用应用等的程序。RAM 22是用作CPU 21的主存储器、工作区域等的存储设备。硬盘23是存储引导程序、各种应用、字体数据、用户文件、电子文档文件等的存储介质。涉及本实施例的程序被存储在硬盘23中,由RAM 22读取,并由CPU 21执行。

[0025] 显示控制器24进行显示器(显示单元)的显示控制。网络控制器25进行与连接在网络中的设备的通信处理的控制。外部存储驱动27针对介质(记录介质) 28,控制数据的读取和写入。键盘控制器29根据键操作、用户对指示设备的操作等,进行操作指示信号的输入处理。

[0026] 图3是示出模块管理单元7管理的设定数据9的结构示例的概念图。设定数据9是作为在例外处理时要调用的函数的登记处理的执行期间,或者在例外处理的执行期间,由模块管理单元7参照或更新的数据结构。各个进程具有一组设定数据9。不特别限定设定数据9的存储区域,只要它具有存储数据的功能即可,诸如文件、存储器或数据库等。设定数据9具有例如第一管理区域和第二管理区域。第一管理区域31是进程信息管理区域,并且第二管理区域32是模块管理单元信息管理区域。

[0027] 第一管理区域31是用于管理关于因程序的执行而正运行的进程3的信息的区域。在信息处理装置1中运行的各个进程,被分配进程ID 33和进程名称34。第一管理区域31存储这两种类型的信息。模块管理单元7参照进程ID 33和进程名称34,能够指定在哪个进程中正在操作该单元。

[0028] 第二管理区域32是用于管理关于在进程3中正运行的模块管理单元7的信息的区域,并且存储关于当进行例外处理时要调用的函数的信息。第二管理区域32具有多个登记函数管理区域。例如,第二管理区域32具有两个区域,即自模块登记函数管理区域(在下文中,被称为自函数管理区域) 35和他模块登记函数管理区域(在下文中,被称为他函数管理区域) 36。

[0029] 自函数管理区域35是如下的区域,其中,存储在自模块(即受模块管理单元7的管理的模块) 请求的例外处理时要调用的函数的信息。另一方面,他函数管理区域36是如下的区域,其中,存储在他模块(即不受模块管理单元7的管理的模块) 已登记在OS 2中的例外处理时要调用的函数的信息。在区域的各个中,存储函数ID 37和函数指针38。函数ID 37表示由OS 2为唯一地识别登记的函数而给出的识别符。函数指针38表示加载登记的函数的存储器的头地址。当例外处理发生时,参照自函数管理区域35和他函数管理区域36的存储信息,进行调用函数的处理。

[0030] 图4示出了OS 2管理的信号通知函数列表10的结构示例。信号通知函数列表10是用于管理当例外发生时、信号被发送到的函数的列表的数据,并且在各个进程中设置一组该数据。不特别限定信号通知函数列表10的存储区域,只要它具有存储数据的功能即可,诸如文件、存储器或数据库等。经由运行时库8,通过在进程3中包括的各个模块或模块管理单

元7与OS 2之间的通信,能够进行信号通知函数列表10的登记或更新。在信号通知函数列表10中,主要管理包括函数ID 37和函数指针38的两种类型的信息。在图4中示出的示例中,登记有共3个函数。当例外处理发生时,从列表的顶部起依次执行函数。即,在这种情况下,首先调用函数ID 37为“001”的第一函数。其被构造为使得当函数的执行完成时,在接下来的控制中,调用函数ID 37为“002”的第二函数,然后,调用第三函数。

[0031] (第一实施例)

[0032] 下面,将描述本发明的第一实施例。将参照图5中示出的流程图,来描述由模块管理单元7进行的一系列处理的示例。将描述在该实施例中如图1所示的在进程3中加载并执行3个模块A、B及C的情况。以下假设加载模块的顺序为如下的顺序,作为他模块的模块A、作为自模块的模块B以及作为自模块的模块C。此外,假设由模块A、模块B和模块C分别定义函数A、函数B和函数C,并且这些函数定义当例外发生时要进行的处理(例如,转储信息获取处理)。此外,由模块管理单元7定义函数D,并且如下的处理被执行,即当例外发生时依次调用要调用的自模块的登记的函数。

[0033] 在图5中,当加载特定模块并且启动模块管理单元7时,模块管理单元7前进到S1。模块管理单元7接收来自模块的请求,以在S1中,登记当例外发生时要调用的函数。此时,模块管理单元7创建设定数据9的新实例,并且将接收到的函数的信息(函数信息)登记在设定数据9中。此时,假设模块管理单元7从其接收到登记请求的模块,受到模块管理单元7的管理。即,该模块是自模块,并且接收到登记函数信息的请求的模块管理单元7,在自函数管理区域35中进行登记。

[0034] 接下来,在S2中,模块管理单元7经由运行时库8,从OS 2获取信号通知函数列表10。然后,在S3中,模块管理单元7确定函数是否已被登记在在S2中获取的信号通知函数列表10中。如果函数已被登记在信号通知函数列表10中,则模块管理单元7前进到S4的处理,而如果函数未被登记,则模块管理单元7移至S6。

[0035] 在S4中,模块管理单元7确定信号通知函数列表10中登记的函数是否为其自身的函数(受其管理的函数)。如果要确定的函数不是其自身的函数,则模块管理单元7前进到S5,登记已被登记在设定数据9中的函数信息,并且前进到S6。此时,已被登记在信号通知函数列表10中的函数,是不受模块管理单元7的管理的他模块的函数。为此,模块管理单元7在他函数管理区域36中进行登记。另一方面,如果模块管理单元7在S4中确定要确定的函数是其自身的函数,则不需要再次登记该函数。因此,处理结束,而没有特别的操作。

[0036] 在S6中,模块管理单元7将其自身的函数,登记在信号通知函数列表10中。在从自模块请求登记的函数的情况下,函数的内容在设定数据9中的自函数管理区域35中存储的函数的列表中。在这种情况下,如果调用其自身的函数,则由CPU 21依次调用并执行在函数的列表中登记的函数。

[0037] 如上所述,每次在进程3中加载模块时,模块管理单元7获取OS 2中登记的信号通知函数列表10。如果存在除其自身的函数之外的登记的函数,则模块管理单元7将函数的列表的信息存储在存储部中,并且代替登记其自身的函数的信息。因此,当在进程3中发生例外时,由模块管理单元7定义的函数被可靠地调用。

[0038] 下面,将参照图6A和图6B详细描述操作。图6A和图6B示出了在第一实施例中当加载模块时,设定数据9和信号通知函数列表10的数据转变的示例。图6A举例说明了针对设定

数据9中的第二管理区域32的数据转变。如由附图标记A-1所示,在初始状态下,在自函数管理区域35和他函数管理区域36二者中没有数据。此外,图6B举例说明了信号通知函数列表10的数据转变。如由附图标记B-1所示,在初始状态下,没有关于函数的列表的登记信息。

[0039] 假设首先加载模块A。模块A是他模块并且不涉及模块管理单元7,因而,在第二管理区域32中的数据中,没有改变发生。假设接下来加载模块B。由于模块B是自模块,所以向模块管理单元7做出用于登记函数B的请求。因此,在图5的S1中,模块管理单元7将函数B的信息,存储在设定数据9的自函数管理区域35中。然后,在图5的S2中,模块管理单元7获取信号通知函数列表10。结果,如由附图标记B-2所示,他模块A的函数A被登记,因而,模块管理单元7从图5的S3前进到S4。在从S4前进到S5之后,如由附图标记A-2所示,模块管理单元7将函数A的信息存储在他函数管理区域36中。在图5的S6中,模块管理单元7将其自身的函数D,登记在信号通知函数列表10中。

[0040] 假设最后加载模块C。模块C是自模块,并且如同在模块B的情况,向模块管理单元7做出用于登记函数C的请求。因此,如由附图标记A-3所示,在图5的S1中,模块管理单元7将函数C的信息,存储在设定数据9中的自函数管理区域35中。此外,虽然如同加载模块B时,在图5的S2中进行获取信号通知函数列表10的处理,但是如由附图标记B-3所示,模块管理单元7的函数D已被登记在其中。因此,如由附图标记A-3所示,不更新他函数管理区域36的信息。

[0041] 参照图6B,当加载模块A时,附图标记B-1的初始状态(没有函数被登记在列表中的状态),转变到附图标记B-2的状态。这是因为,模块A是他模块,因而被登记在信号通知函数列表10中,而不涉及模块管理单元7。因此,函数A的信息被登记在信号通知函数列表10中。

[0042] 接下来,当加载模块B时,在图5的S6中,模块管理单元7更新信号通知函数列表10的信息。因此,如由附图标记B-3所示,模块管理单元7的函数D被登记在信号通知函数列表10中。当最后加载模块C时,如同当加载模块B时,模块管理单元7参照信号通知函数列表10的信息,然而,登记的函数已被模块管理单元7本身登记。因此,不更新信号通知函数列表10,并且维持由附图标记B-3所示的状态。

[0043] 以上已在图6A和图6B中描述了当依次加载模块A、B及C时的数据转变。同时,当卸载模块时进行的处理,是与当加载模块时逆转的数据转变。例如,假设如下的状态,即如由附图标记A-3和B-3所示,加载了所有的模块A、B及C。如果在这种状态下卸载模块C,则第二管理区域32从附图标记A-3的状态,转变到附图标记A-2的状态。此时,信号通知函数列表10从附图标记B-3的状态,转变到附图标记B-2的状态10。

[0044] 接下来,将参照图7描述本实施例的例外处理。图7是示出当例外发生时,由模块管理单元7进行的例外处理的示例的流程图。由运行时库8在进程3中检测例外的发生。当检测到例外的发生时,OS 2向信号通知函数列表10中登记的函数,依次发送信号。如果模块管理单元7的函数被登记在信号通知函数列表10中,则在S11中,模块管理单元7接收由OS 2发送的信号。接下来,在S12中,执行登记的函数的处理。在本实施例中,调用由模块管理单元7定义的函数D。函数D具有依次调用在自函数管理区域35中登记的函数的列表的处理的内容。因此,在本实施例中,如由附图标记A-3所示,按顺序执行函数B和函数C。在S12完成之后,模块管理单元7前进到S13的处理,并且确定函数是否被登记在设定数据9中的他函数管理区域36中。如果函数被登记在他函数管理区域36中,则模块管理单元7前进到S14,而如果没有

函数被登记在他函数管理区域36中,则单元移至S15。

[0045] 在S14中,模块管理单元7调用在他函数管理区域36中登记的函数。此时,与在S11中接收到的信号相同的信号,被发送给该函数。在本实施例中,如由附图标记A-3所示,执行函数A的处理。然后,模块管理单元前进到S15。在S15中,进行删除设定数据9的处理。进行删除设定数据9的处理,以防止数据残留在存储器中,或者在进程3的终止之后,作为文件残留。在S15的处理之后,运行时库8执行终止进程3的处理。

[0046] 当如上所述,在进程中加载特定模块时,模块管理单元7将针对其进行信号接收登记的函数的信息,存储在存储区域中,并且将受模块管理单元的管理的函数,覆写在其上。因此,当在进程中发生例外时,能够控制函数的调用,在例外的发生期间,能够将相同的信号通知给所有的函数,并且能够获取在进程的异常终止时的转储信息。

[0047] 在本实施例中,区分并管理受管理的第一类型的模块(自模块)和不受管理的第二类型的模块(他模块)。如果第二类型的模块的函数被登记在例外处理中,则执行将该函数的信息存储在存储部中的处理,并且进行使第一类型的函数的处理执行的函数登记,而非第二函数。在函数登记中,例如,在函数的列表中描述了进行如下的处理的函数,即调用第一类型的模块的函数或第一类型的模块的函数的信息。

[0048] 在本实施例中,举例说明了在进程中仅存在1个模块管理单元的情况,以描述在模块加载和例外发生时,模块管理单元的处理和数据转变的操作。当加载模块时,模块管理单元7监视信号通知函数列表10,以确定他模块的函数是否被登记。当他模块的函数被登记在信号通知函数列表10中时,通过利用用于执行其自身的模块的函数的处理的函数来替换该函数,从而当例外发生时,调用模块管理单元7的函数。此外,由于通过在设定数据9中区分自模块和他模块的信息的管理区域,来管理自模块和他模块的信息,所以能够在自模块的函数被调用之后,调用他模块的函数。因此,不仅能够在自模块中,还能够在他模块中,进行例外处理。

[0049] (第二实施例)

[0050] 接下来,将描述本发明的第二实施例的信息处理装置的结构、处理等。在本实施例中,将举例说明在1个进程中存在多个模块管理单元的情况。应当注意,通过使用上面对与第一实施例相同的构成要素而使用的附图标记,将省略其详细描述,并且将主要描述不同之处。这种省略描述的方式也适用于下面将要描述的另一实施例。

[0051] 图8是示出本实施例的信息处理装置的结构示例的框图。与第一实施例的不同之处是:模块管理单元7在1个进程3中存在为多个实例。在图8中,举例说明了模块管理单元7A(模块管理单元A)和模块管理单元7B(模块管理单元B)。假设模块管理单元A管理模块B(模块5)和模块C(模块6),并且模块管理单元B管理模块D(模块11)和模块E(模块12)。此外,当在1个进程3中存在多个模块管理单元时,存在设定数据9的1个实例。在独立于模块管理单元中的各个来管理设定数据9的方法的情况下,在例外发生时、调用由模块管理单元中的各个管理的模块组的顺序是独立的,因而无法控制该顺序。为此,一组设定数据9被构造为由多个模块管理单元共享。

[0052] 由于在图8的结构中存在一组设定数据9,因此当模块管理单元B创建新的设定数据9时,即使模块管理单元A已经运行,但是例如,包括在现有的设定数据9中的信息丢失。在这种情况下,存在当例外发生时,某些函数不被调用的可能性。此外,当卸载模块管理单元B

时,例如,如果删除设定数据9,即使模块管理单元A正在运行,但是包括在设定数据9中的信息丢失。因此,在本实施例中,将参照图9和图10描述当在1个进程3中存在多个模块管理单元7时,设定数据9的更新和删除处理。

[0053] 图9是示出由模块管理单元7进行的函数信息登记处理的示例的流程图。图9中示出的处理涉及图5的S1的处理,并且当在进程3中加载特定模块时,由例如模块管理单元7A执行图9中示出的处理。首先在S21中,模块管理单元7A确定设定数据9是否已存在。设定数据9已存在的事实是指,已加载了其他模块管理单元7B并且创建了设定数据9的事实。因此,如果设定数据9已存在,则模块管理单元前进到S23,而不创建新的设定数据9。另一方面,如果没有设定数据9,则模块管理单元7A将本身认作第一加载的模块管理单元,并移至S22。在S22中,第一加载的模块管理单元7A创建新的设定数据9。然后,模块管理单元前进到S23。

[0054] 在S23中,模块管理单元7A将其自身的条目信息(entry information)登记在设定数据9中。条目信息是指与进程3中加载的模块管理单元的实例相关的信息。使用这样的条目信息,使得多个模块管理单元中的第一模块管理单元,能够确定第二模块管理单元的有无,换言之,是否已存在正在运行的第二模块管理单元。此外,作为用于确定已将函数登记在自函数管理区域35的模块管理单元的信息,条目信息也是必要的。作为模块管理单元的条目信息,例如,模块ID和模块名称被登记在设定数据9中。然后,在S24中,模块管理单元7A将从自模块请求登记的函数的信息,添加到设定数据9中的自函数管理区域35中。此时,为了识别已设置信息的模块管理单元,进行告知(impart)模块管理单元的条目信息的处理。

[0055] 接下来,将在图10中描述在本实施例中,由模块管理单元进行的删除设定数据9的处理的示例。图10的流程图中示出的处理涉及图7的S15的处理,并且当在进程3中发生例外时,由模块管理单元7执行图10的流程图中示出的处理。首先,在S31中,模块管理单元7A例如参照在设定数据9中登记的模块管理单元的条目信息。模块管理单元7A确定在设定数据中是否存在其他模块管理单元7B的条目信息。存在其他模块管理单元7B的条目信息,是指仍在加载其他模块管理单元7B的状态。因此,在这种情况下,模块管理单元7A前进到S33,而不删除设定数据。在S33中,模块管理单元7A删除其自身的条目信息。另一方面,如果没有其他模块管理单元7B的条目信息,则除模块管理单元7A之外,没有加载的模块管理单元。因此,模块管理单元7A从S31移至S32,并且删除整个设定数据9。

[0056] 如上所述,当在本实施例中,在同一进程中存在多个模块管理单元并且参照一组设定数据9时,确定其他模块管理单元是否正在运行。根据该确定的结果,执行设定数据9的新创建、删除等的处理。根据本实施例,由于因设定数据9的覆写或删除而不存在丢失登记的函数的信息的可能性,所以能够适当地执行例外处理。

[0057] (第三实施例)

[0058] 接下来,将描述第三实施例的信息处理装置的处理。在第一实施例中,已经描述了当在进程中发生例外时,调用登记的函数的处理。在此,假设如下的情况作为示例,即在当例外发生时、存储器容量不足的状态下,要调用函数。在这种情况下,在当例如函数的调用发生过多次数等时,从而发生堆栈溢出的情况下,存在进程被终止而没有调用所有的函数的可能性。在这种情况下,令人担忧的是,无法获取更重要的模块的转储信息,而仅获取他模块的转储信息。因此,在第三实施例中,将描述当例如,诸如个人计算机(PC)等的信息处理装置相对于资源处于困难的状况时,依次调用具有更高级别的重要性的函数的处理。

[0059] 将参照图11的流程图,描述在本实施例中当例外发生时,由模块管理单元7进行的处理的示例。在图11中,在从S11至S15的步骤中示出的处理,与在第一实施例中的图7中所描述的相同。因此,将省略针对从S11至S15的步骤的详细描述,并且将主要描述从S41至S44的不同的处理。

[0060] 首先,当运行时库8检测到例外时,OS 2发送信号,并且模块管理单元7在S11中接收信号,并且处理前进到S41。在S41中,模块管理单元7获取信息处理装置1的资源信息。资源信息是指例如CPU 21、RAM 22、硬盘23等的使用率。接下来,在S42中,模块管理单元7将在S41中获取的资源信息的值与预定的阈值进行比较。根据所有的函数都不可能调用的资源状况,来设置该阈值。该设置的方法包括预先设置固定值的方法,以及设置根据登记的函数的数量等而动态地决定的可变值的方法。在S42中进行如下的处理,即确定资源信息的值是否超过阈值,并且如果该值超过阈值,则处理前进到S43,而如果该值不超过阈值,则处理移至S12。

[0061] 在S43中,模块管理单元7获取各个模块的重要性的级别的信息。具体而言,假设为模块设置重要性的级别,并且在该步骤中,模块管理单元7例如参照表示各个模块的重要性的级别的值。然后,在S44中,模块管理单元7更新设定数据9,并且通过以重要性的级别的降序来重新排列在自函数管理区域35中登记的函数,从而改变调用顺序。例如,如果随着表示重要性的级别的值越来越高,模块的重要性的级别增加,则以表示重要性的级别的值的降序,来进行整理函数的列表的处理。然后,处理前进到S12以及随后的步骤。由于在S12中,以函数的列表的顺序执行函数的处理,所以从具有最高的重要性的级别(即优先级)的函数起,依次调用函数。

[0062] 在本实施例中,通过根据当例外发生时信息处理装置的资源状况(诸如困难的程度或充足的程度),来控制调用函数的顺序,能够优先获取具有更高的重要性的级别的模块的信息。用于设置重要性的级别的标准包括具有在开发阶段中发生漏洞或错误的更高的可能性的模块的顺序,以及具有更大的量级的模块的顺序。因此,即使无法获取完整的转储信息,也能够提高获取用于分析例外的原因的有用信息的可能性。

[0063] (其他实施例)

[0064] 还可以通过读出并执行记录在存储介质(也可更完整地称为“非暂时性计算机可读存储介质”)上的计算机可执行指令(例如,一个或多个程序)以执行上述实施例中的一个或多个的功能、并且/或者包括用于执行上述实施例中的一个或多个的功能的一个或多个电路(例如,专用集成电路(ASIC))的系统或装置的计算机,来实现本发明的实施例,并且,可以利用通过由系统或装置的计算机例如读出并执行来自存储介质的计算机可执行指令以执行上述实施例中的一个或多个的功能、并且/或者控制一个或多个电路以执行上述实施例中的一个或多个的功能的方法,来实现本发明的实施例。计算机可以包括一个或多个处理器(例如,中央处理单元(CPU),微处理单元(MPU)),并且可以包括分开的计算机或分开的处理器的网络,以读出并执行计算机可执行指令。计算机可执行指令可以例如从网络或存储介质被提供给计算机。存储介质可以包括例如硬盘、随机存取存储器(RAM)、只读存储器(ROM)、分布式计算系统的存储器、光盘(诸如压缩光盘(CD)、数字通用光盘(DVD)或蓝光光盘(BD)TM、闪存装置以及存储卡等中的一个或多个。

[0065] 本发明的实施例还可以通过如下的方法来实现,即,通过网络或者各种存储介质

将执行上述实施例的功能的软件(程序)提供给系统或装置,该系统或装置的计算机或是中央处理单元(CPU)、微处理单元(MPU)读出并执行程序的方法。

[0066] 虽然参照示例性实施例对本发明进行了描述,但是应当理解,本发明不限于所公开的示例性实施例。应当对所附权利要求的范围给予最宽的解释,以使其涵盖所有这些变型例以及等同的结构和功能。

[0067] 本申请要求2015年3月17日提交的日本专利申请第2015-053658号的优先权,该申请的全部内容通过引用并入本文。

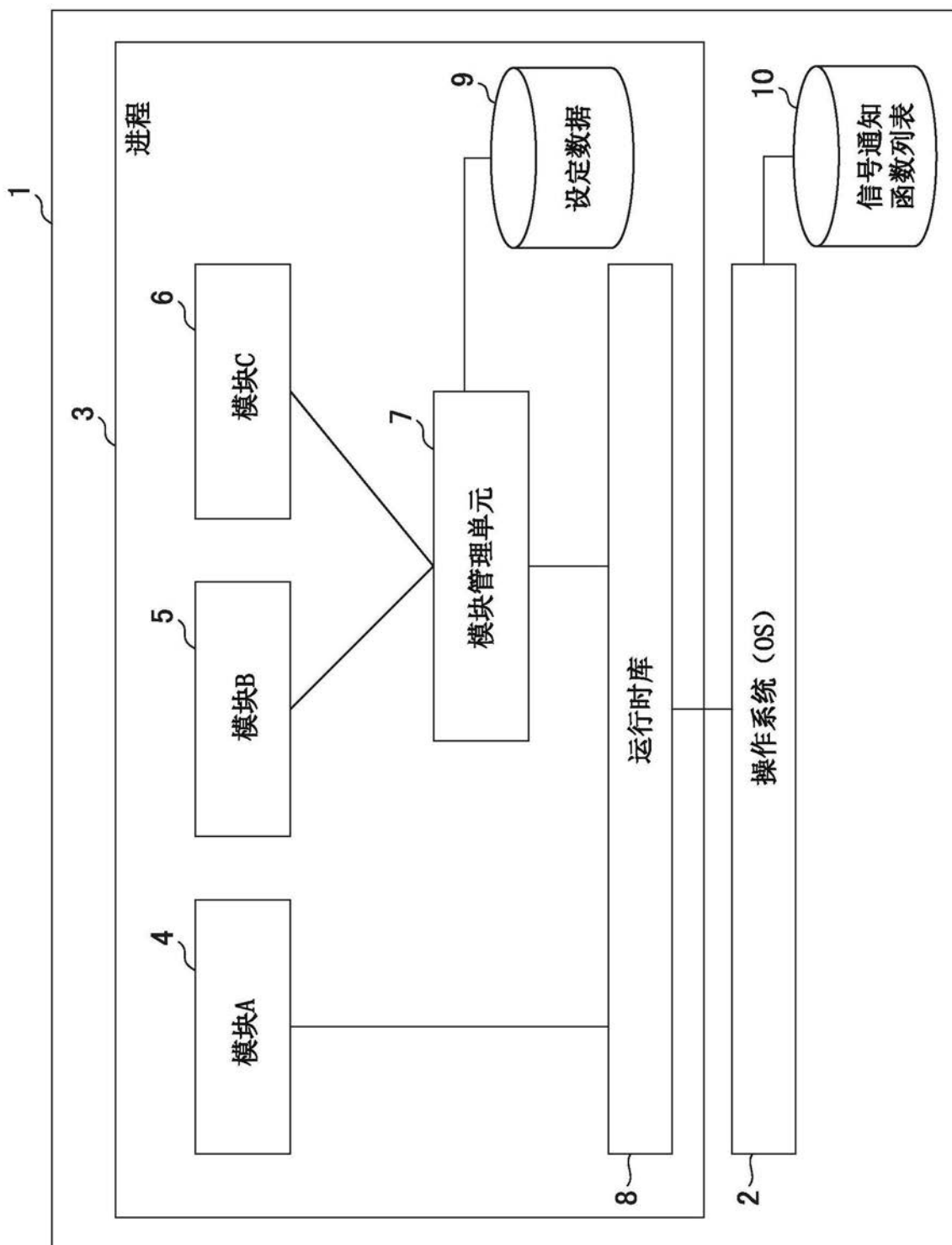


图1

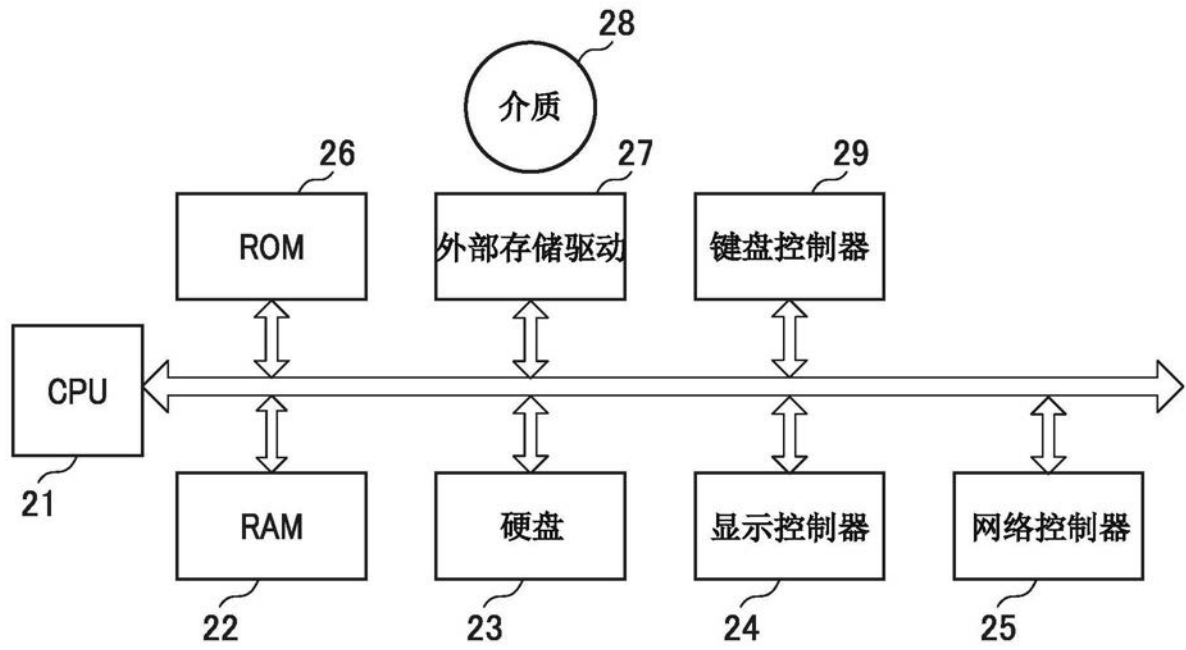


图2

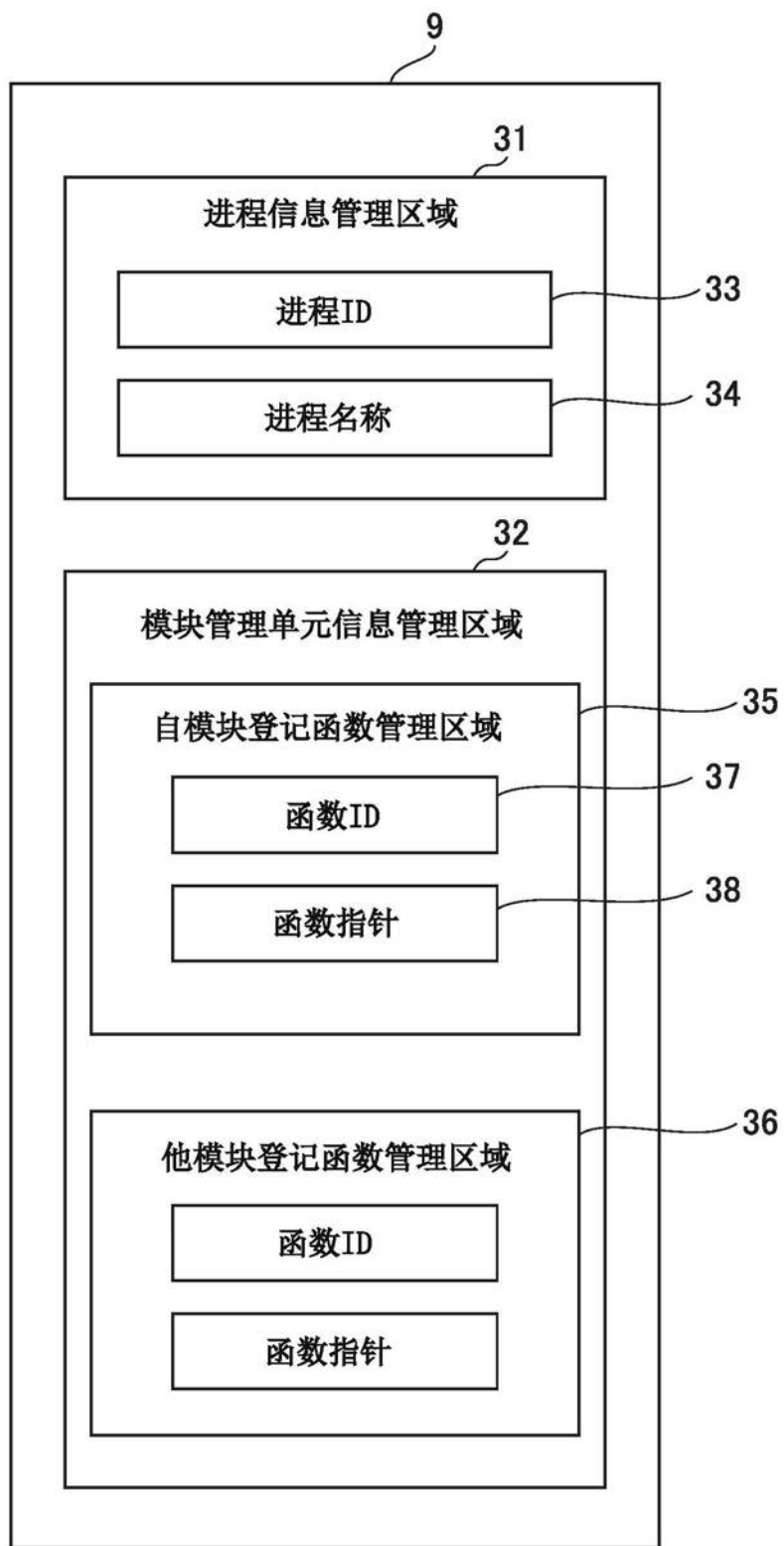


图3

10

37		38
函数ID	函数指针	
001	0x40e7fe	
002	0x47c93c	
003	0x43b83a	

图4

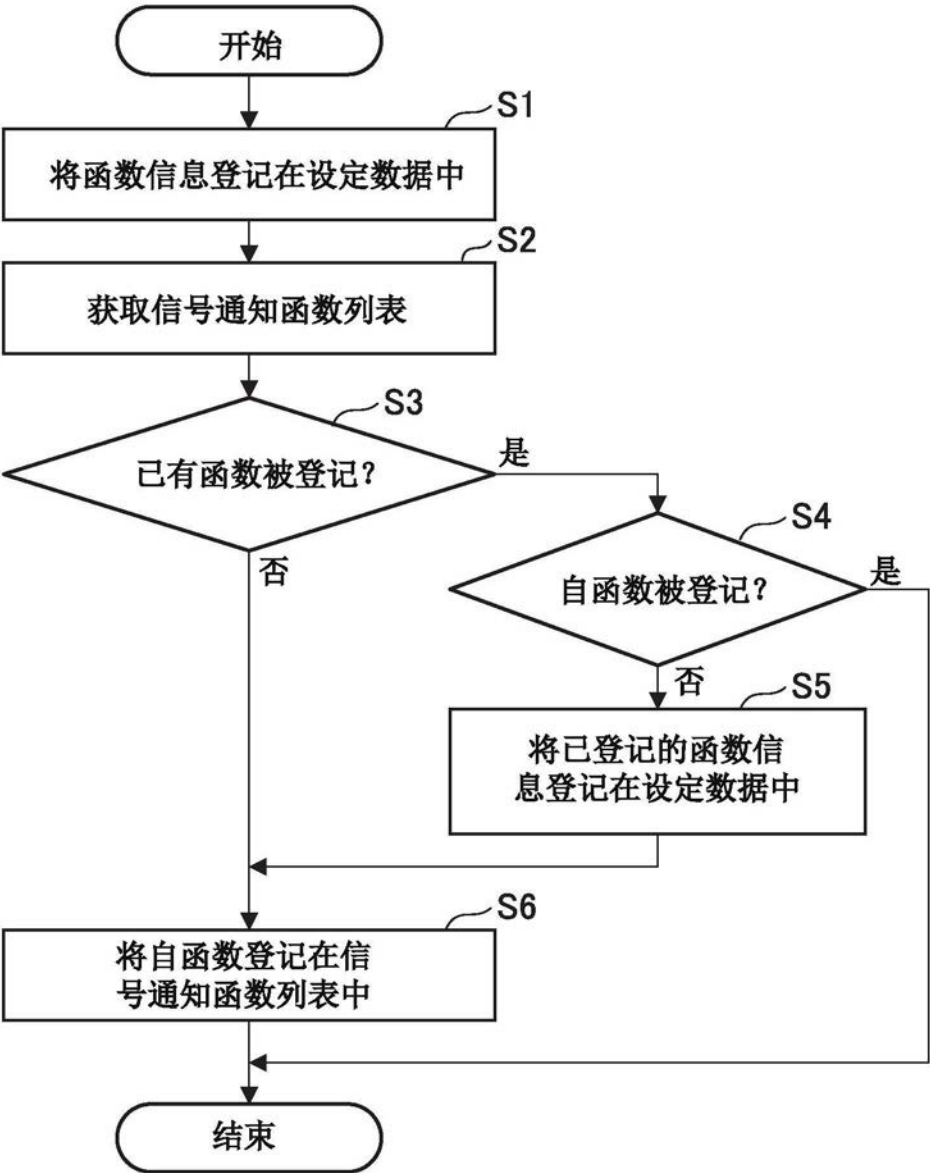


图5

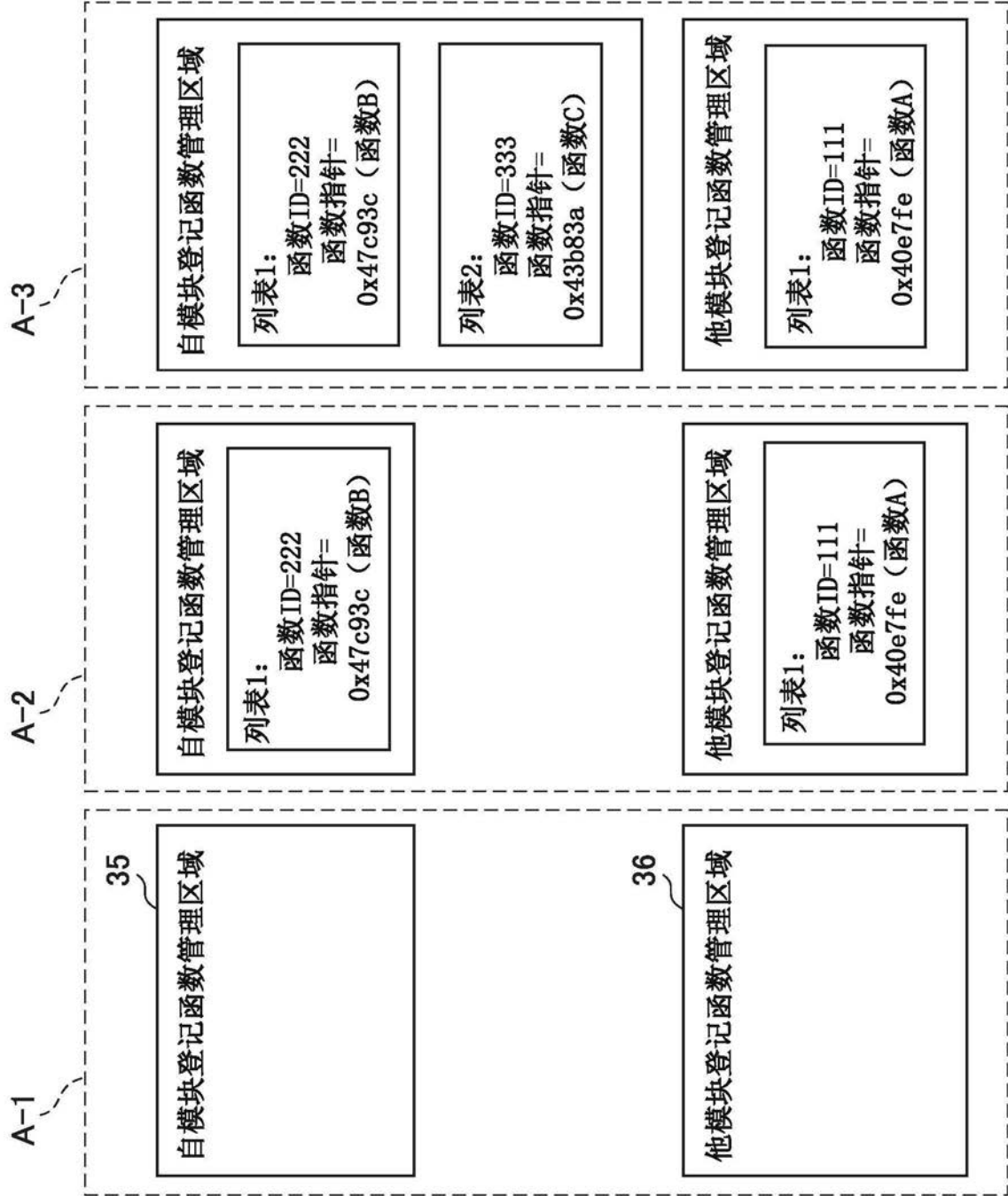


图6A

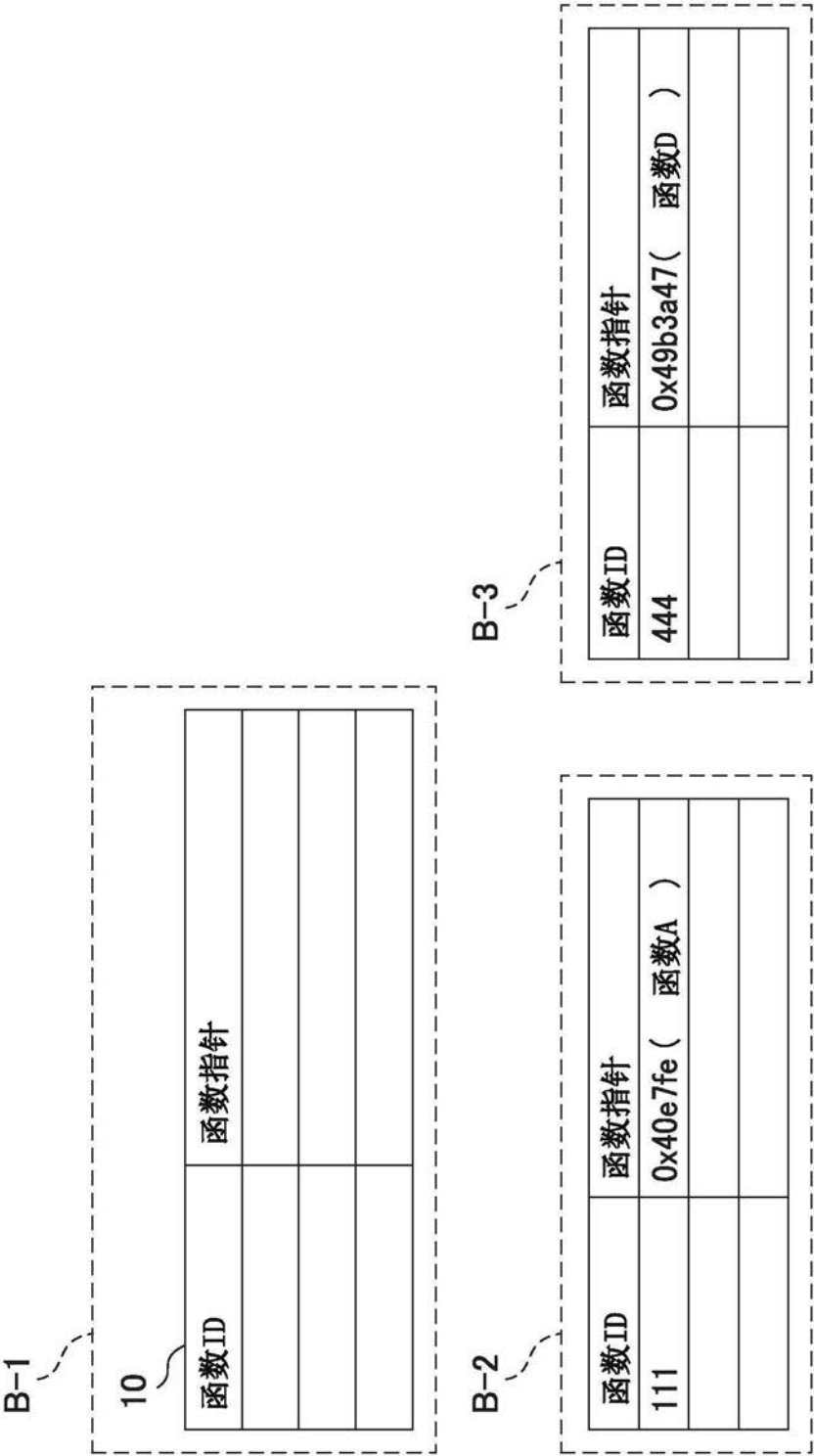


图6B

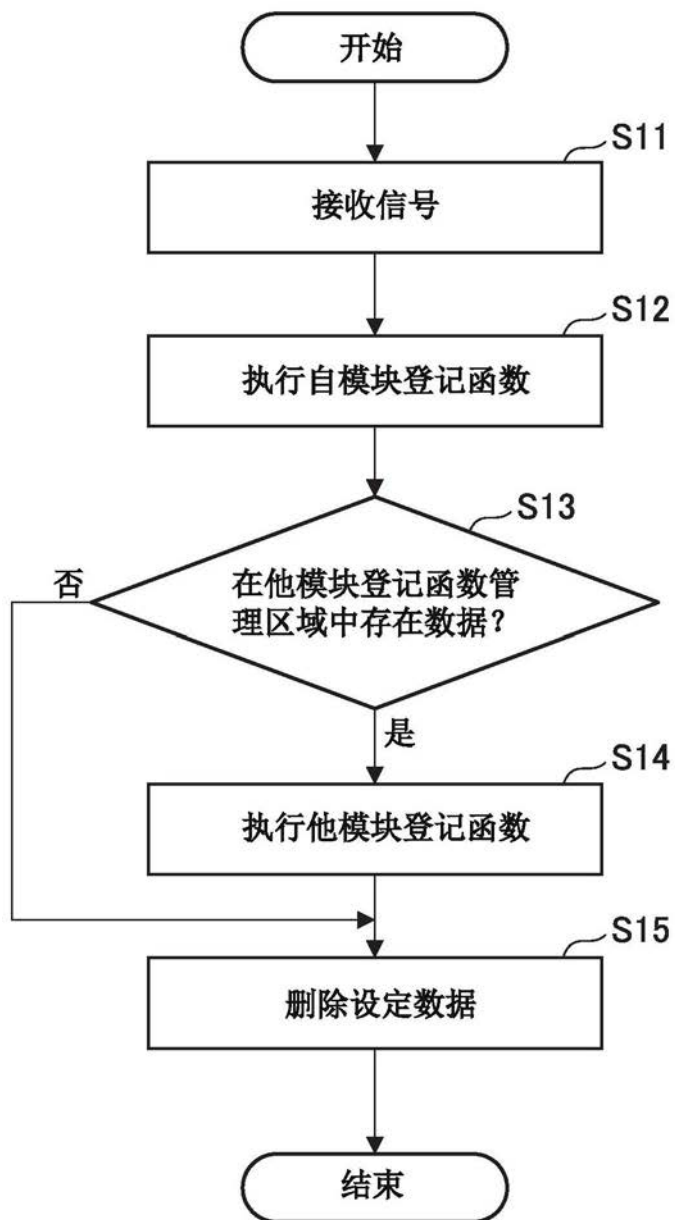


图7

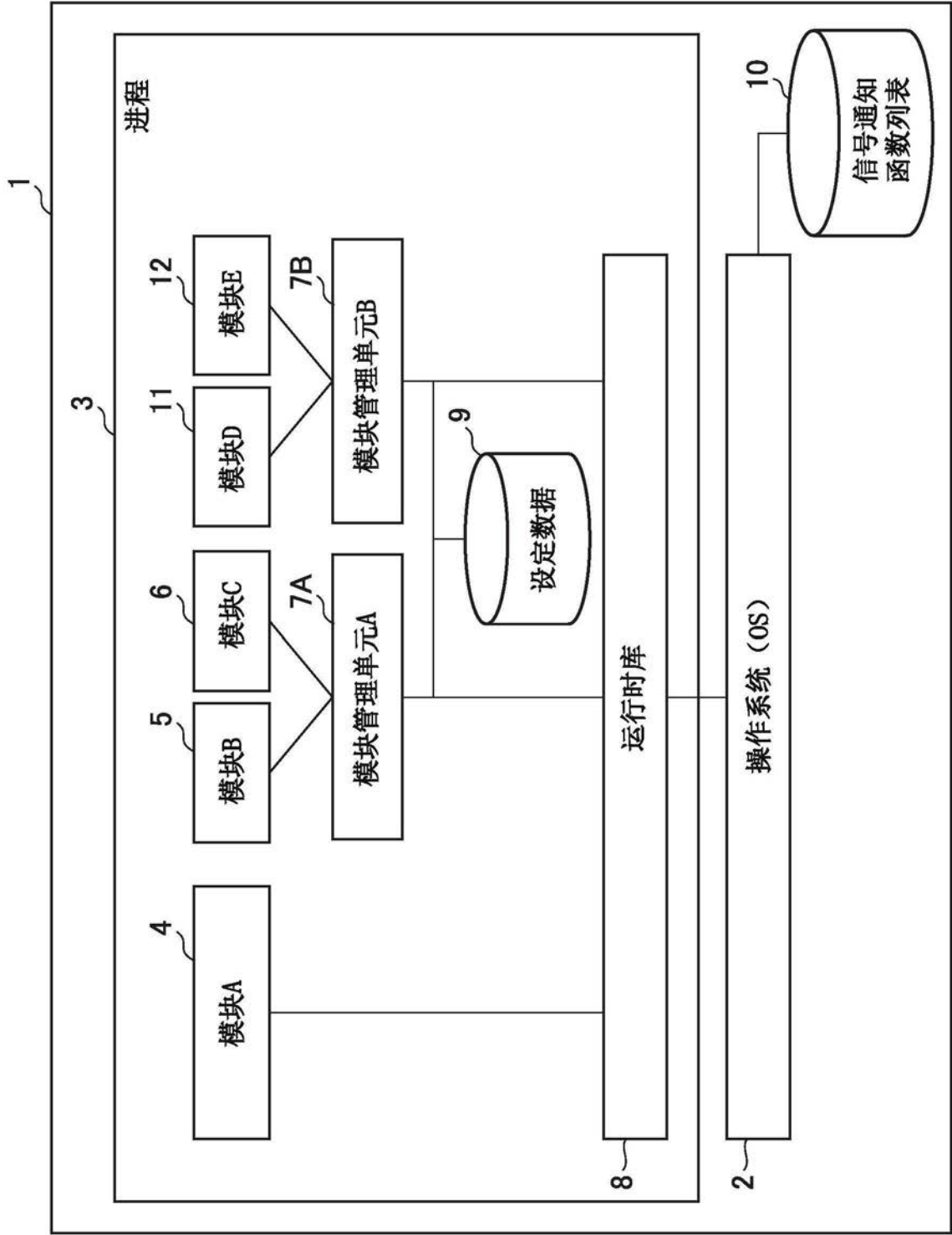


图8

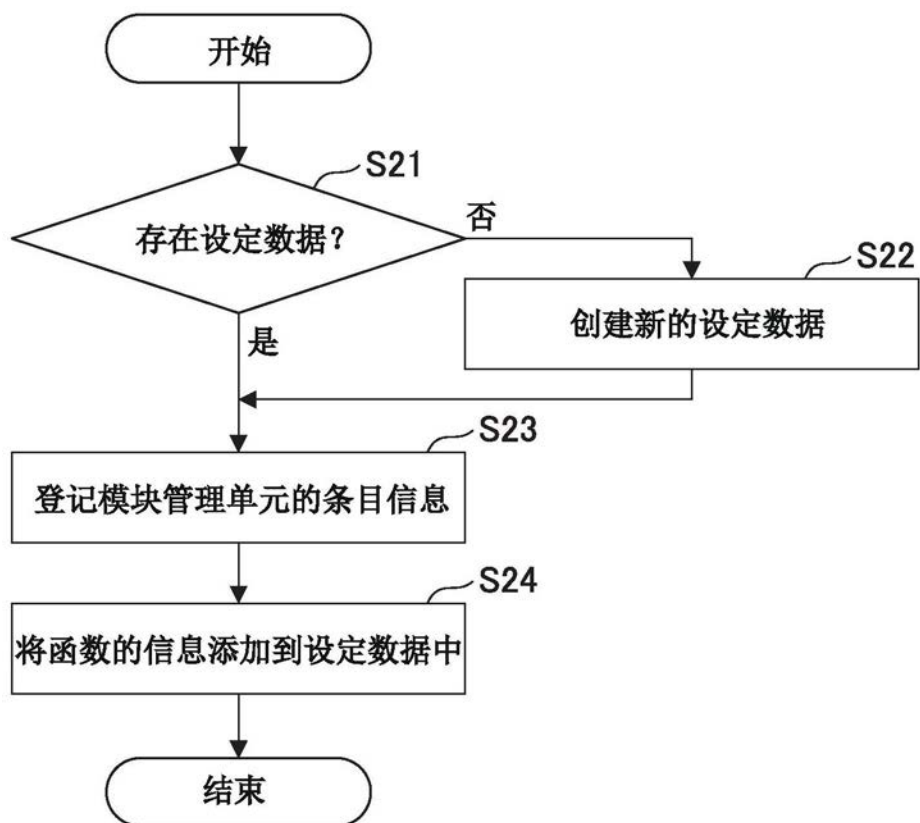


图9

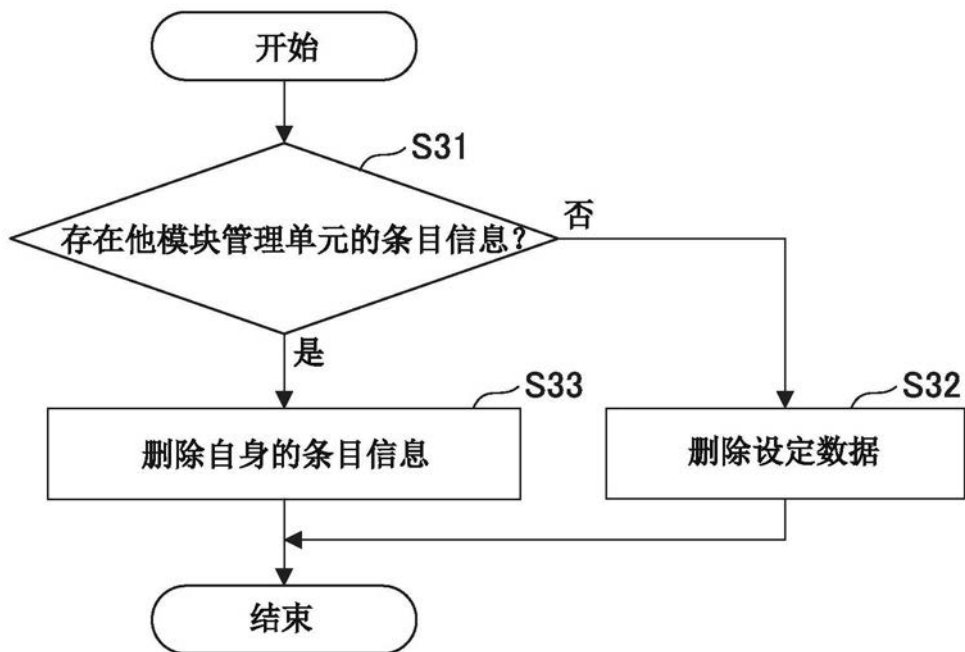


图10

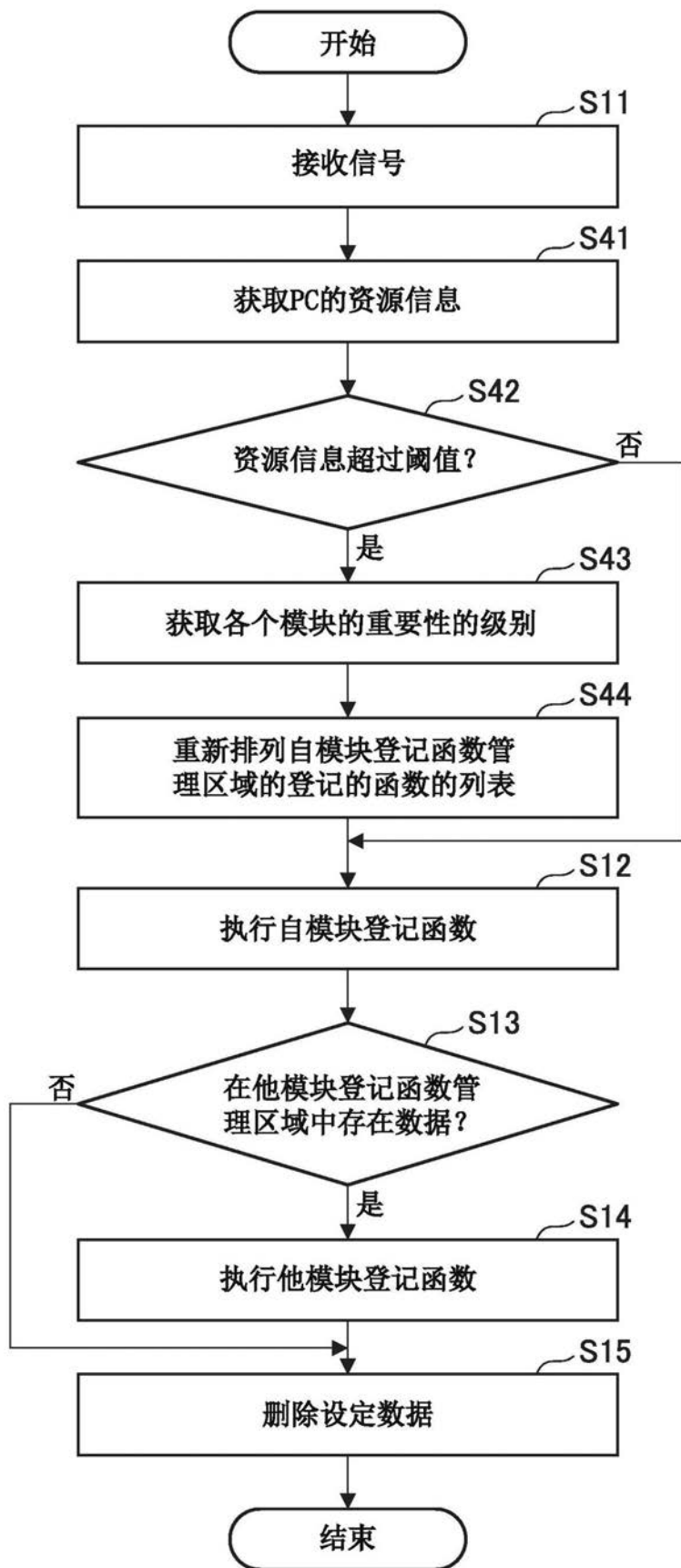


图11