[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 9/44 (2006.01)
G06F 9/46 (2006.01)

[12] 发明专利说明书

专利号 ZL 200410088292.5

[45] 授权公告日 2008 年 10 月 15 日

[11] 授权公告号 CN 100426224C

[22] 申请日 2004.10.21
[21] 申请号 200410088292.5
[30] 优先权
[32] 2003.10.24 [33] US [31] 10/693,749
[73] 专利权人 微软公司
地址 美国华盛顿州
[72] 发明人 K·D·雷 M·佩纳多
P·英格兰德 T·V·库里恩
[74] 专利代理机构 上海专利商标事务所有限公司
代理人 陈 斌
[56] 参考文献
CN1444742A 2003.9.24
A TRUSTED OPEN PLATFORM. PAUL ENGLAND, BUTLER LAMPSON, JOHNMANFERDELLI, MARCUS PEINADO, BRYAN WILLMAN. IEEE. 2003
审查员 孟宪超

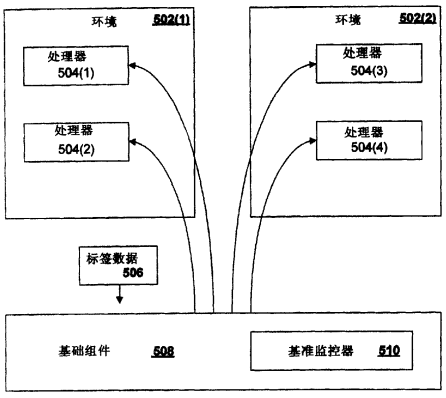
权利要求书 2 页 说明书 17 页 附图 6 页

[54] 发明名称

用于应用程序的划分的方法和系统

[57] 摘要

应用程序分解或划分用于将安全特征集成到常规的应用程序中。 应用程序的功能依照给定的行动是否涉及敏感数据的处理被划分成两个组。 创建单独的软件对象(处理器)来执行这两组行动。 可信处理器处理安全数据并运行在高保证环境中。 当另一处理器遇到安全数据时，该数据被发送到可信处理器。 以允许数据被路由到可信处理器的方式包装该数据，并防止数据被不同于可信处理器的任一实体破译。 提供了一种基础结构，它包装对象、将它们路由到正确的处理器，并允许通过反向通往已知为可信的基础组件的可信链来证明它们的完整性。



1．一种用于在第一操作系统环境中执行的、由第一软件对象处理应用有一保证策略的数据的方法，其特征在于，所述方法包括：

所述第一软件对象遇到所述数据；

所述第一软件对象确定所述数据无法由所述第一软件对象处理；

如果所述数据无法由所述第一软件对象处理，所述第一软件对象促使所述数据被提供给在第二操作系统环境中执行的第二软件对象，所述第二操作系统环境提供所述第二操作系统环境中执行的动作将被正确执行的第一保证级别，其中，所述第二软件对象通过一方式处理所述数据，该方式使用所述保证策略来创建对所述数据的篡改的阻止，其中对所述数据的篡改是由发生在所述第二操作系统环境外部的动作引起的。

2．如权利要求 1 所述的方法，其特征在于，对篡改的阻止包括对所述数据的改变的阻止。

3．如权利要求 2 所述的方法，其特征在于，所述数据在一可视显示设备上显示，并且其中，对篡改的阻止包括在所述可视显示设备的位置上显示所述数据的表示，并取代不同于在所述位置上呈现的所述表示的任何图像。

4．如权利要求 1 所述的方法，其特征在于，所述第一软件对象促使所述数据的表示在一可视显示设备上显示，所述表示包括一个或多个不可破译的图形。

5．如权利要求 4 所述的方法，其特征在于，所述表示：彼此大小相同，或具有与所述数据的内容不相关的大小。

6．如权利要求 4 所述的方法，其特征在于，所述第一软件对象或所述第二软件对象、或所述第一软件对象和所述第二软件对象的组合，促使所述可视显示设备上显示的项目在至少一个方面变化，以准许察看由所述第二软件对象生成的所述数据的图像。

7．如权利要求 2 所述的方法，其特征在于，所述数据使用键盘来提供，并且其中，对篡改的阻止包括阻止对从所述键盘转移到所述第二软件对象的输入流的所述数据的改变。

8．如权利要求 1 所述的方法，其特征在于，所述策略指定所述数据由所述第二软件对象处理。

9. 如权利要求 1 所述的方法，其特征在于，所述数据包括将所述第二操作系统环境标识为处理所述数据的位置的第一标签，或与该第一标签相关联。

10. 如权利要求 9 所述的方法，其特征在于，所述数据包括将所述第二软件对象标识为所述数据的处理器的第二标签，或与该第二标签相关联，并且其中，所述第二操作系统环境基于所述第二标签将所述数据路由到所述第二软件对象。

11. 如权利要求 1 所述的方法，其特征在于，所述第二操作系统环境与描述所述第二操作系统环境的行为的第一规范相关联，并且其中，所述保证策略提供所述第二操作系统环境将符合所述规范。

12. 如权利要求 1 所述的方法，其特征在于，所述第一操作系统环境与描述所述第一操作系统环境行为的第二规范相关联，并且其中，所述第一操作系统环境提供所述第一操作系统环境中执行的行动将被正确执行的第二保证级别，所述第二保证级别相对低于所述第一保证级别。

用于应用程序的划分的方法和系统

技术领域

本发明一般涉及计算领域，尤其是提供了一种以允许需要可信或安全测量的操作集成到普通、非安全软件中的方式支持应用程序的划分和分解的机制。

背景技术

在计算领域，在一方面提供高度安全的系统和另一方面提供大量功能性特征和高度可扩充性的系统之间存在一种紧张关系。计算领域的安全性取决于用高度的确定性理解并预测计算机系统的行为（即，软件和硬件的行为）的能力－即，确保系统不会通过无意的误用或故意的攻击以不同于为其设计的方式表现。例如，被设计成保护有版权的资料不被复制的计算机系统仅在可确保系统实际上做设计它所做的事情的方面是可信的。然而，大型、开放的体系结构往往是难处理且复杂的，从而很难分析其行为，因为有大量的变量可影响该行为。当前，诸如全服务（full-service）操作系统或文字处理器等大型复杂的程序看似不可能将其行为核实到高度的确定性。可能书写可在各种各样的条件和攻击种类下测试并核实其行为的小程序，但是这一程序只能执行有限的功能组。由此，在提供大量的功能和提供高度的安全性之间存在一种紧张关系。

提出的一种解决方案是并肩运行两个系统－一个具有高度功能性的大系统，以及另一具有高度安全性的小系统。由此，诸如 WINDOWS XP 等全服务操作系统可以连同一小型、高保证操作系统一起运行。只要在全服务操作系统中出现需要以具有高度可信的紧密控制的方式执行的事件，就可将该任务传递到高保证操作系统。

操作系统提供了其它程序可在其中执行的环境。然而，两个操作系统可并肩存在这一纯粹的事实无法解决给定的应用程序如何利用两个环境的问题。期望应用程序能够使用全特征环境来执行大多数功能（即，不需要高度安全性的功能），并使用高保证环境来执行的确需要高度安全性的功能。此外，期望以提供集成的用户体验的方式来使用这两个环境。

鉴于上述原因，需要一种克服现有技术的缺点的系统。

发明内容

本发明提供了一种应用程序的功能可被分解（factor）或划分成多个部分的机制－即，需要某一程度的安全或保护的行动，以及不需要那些安全或保护的行动。依照本发明，应用程序被实施为至少两个软件对象：运行在全特征（但是低保证）环境中的软件对象，以及运行在高保证（但特征有限）环境中的另一软件对象。当全特征环境中的对象执行时，它可能遇到需要某一程度的保护的数据（如，该数据可能需要保密，或它可能需要在确定该数据未被篡改的意义上是可核实的）。当运行在全特征环境中的软件对象遇到这样的数据时，该软件对象促使该数据被传递到高保证环境。在高保证环境中操作的软件对象然后操作该数据。当处理该数据时需要的数据的任何输入、输出或存储使用高保证环境来执行，以保护该数据不被该高保证环境外部出现的事件截取或篡改。

这两个环境由一提供两个环境进行通信所需要的基础结构（或“管道（plumbing）”）的基础组件主管（host）。例如，需要在高保证环境中处理的数据对象可具有该基础组件生成的包装。该包装可标识向其路由该数据对象用于处理的环境，并且也可提供能够核实该数据对象自从被该基础组件包装以来未被修改的事实的封印。由此，软件对象可将数据对象传递到基础组件，基础组件能够：（1）确定该数据对象应当被传递到哪一环境，以及（2）核实该数据对象自从创建以来未被篡改。后一行动提供了允许跨平台建立对象的可信性的基础结构：如果在第一机器上（具有第一基础组件）创建了对象，则该第一基础组件签署该对象为该对象是在该基础组件已知的高保证环境中生成的确切对象的证明。当然后在另一机器上打开该对象时，另一机器然后可以核实该签名，并判定它们是否信任签署者。（如，某些机器和/或基础组件可能比其它机器更善于确保其主管的环境的正确行为；每一机器可为其自己决定它是否信任创建给定数据对象的平台。）

下文将描述本发明的其它特征。

附图说明

当结合附图阅读时，可以更好地理解以上概述以及以下较佳实施例的详细描述。为说明本发明的目的，附图中示出了本发明的示例性构造；然而，本发明不限

于所揭示的具体方法和手段。附图中：

图 1 是可实现本发明的各方面的示例计算环境的框图；

图 2 是被分解成组成功能的应用程序的框图；

图 3 所示是将数据路由到应用程序的不同组件的框图；

图 4 是可分解应用程序的用户界面的示例的框图；

图 5 是支持分解的应用程序的使用的示例性体系结构的框图；

图 6 是可使用分解的应用程序的环境的示例层次结构的框图；

图 7 是在支持分解的应用程序的使用的环境中使用的示例数据对象的框图；

图 8 是可通过其在分解的应用程序中处理数据的示例过程的流程图。

具体实施方式

综述

本发明提供了一种允许应用程序被划分或“分解”成安全和非安全组件，并允许这些组件共同工作以提供关于该应用程序的集成用户体验的机制。例如，文字处理程序可被划分成一执行大多数布局、编辑、打印、拼写检查、语法检查等与文字处理器关联的功能的非安全组件，以及一启用需要某一保护方式的数据对象的显示和编辑的安全组件。非安全组件可运行在普通、开放环境中，如典型的商业操作系统。安全组件可运行在允许某些类型的软件以该软件将正确表现的高保证运行的高保证环境中。本发明提供了关于这一情形的各种特征。首先，本发明提供了一种跨两个组件的用户体验，使得从用户的观点来看，用户看似尽可能地使用单个应用程序。第二，本发明提供允许应用程序处理安全和非安全数据的基础结构或“管道”，并用于跨分区使用的这类数据。

对于用户体验，一般情况是用户开始使用应用程序的非安全部分。在这一使用期间出现的数据（如，文字处理文档中的敏感文本项目、电子表格中的敏感金融数字等）由应用程序的非安全部分以某一方式较佳地集成到用户体验中，即使该数据实际上无法由该非安全部分显示。例如，如果文字处理文档中有敏感或机密文本项目，则文字处理器的非安全部分能够显示标识该项目的框，以及表示即使无法显示该文本也存在的事实的某一类型的图形（如，指示文本的弯曲的线条）。在一个示例中，用户可在该框或图形上点击，然后可调用应用程序的安全部分以在出现该框的屏幕的同一位置显示文本。由此，对于用户体验集成了应用程序的安全和非安

全部分。

对于基础结构，本发明提供了一种允许一个环境中的数据对象被路由到另一环境的机制。通常，机密或敏感的数据对象一并由此需要由安全部分处理一将被加密，使得非安全部分无法读取它。由此，这一对象通常由在生成该数据对象中起作用的每一组件包装在一系列包装中。每一包装较佳地包含附加该包装的组件的标识符，以及允许核实该包装内的数据的完整性的封印。外部包装较佳地由创建该包装的机器上的可信根附加一即，在单个机器上主管各种环境，并由某一公知的授权机构核实为其行为可信的基础组件。该外部包装允许该基础组件担当路由器的作用。由此，当非安全部分遇到该对象时，它可能无法读取该对象，但是可将该对象标识为要发送到另一环境用于处理的对象。由此，非安全部分将该对象发送到基础组件，它然后基于包装中标识了哪一环境将该对象路由到正确的环境。

另外，如果对象在第一机器上创建，并在第二机器上打开，则第二机器可通过检查作为外部包装的一部分的签名确定该对象的可信性。应当理解，当基础组件包装并密封对象时，基础组件本质上证明在对象的创建过程中，基础组件在保护创建过程不受外部篡改的方面正确地执行了其功能。（基础组件通常提供允许高保证环境保护其自身不受篡改的机制一如，可信处理器模块（TPM）、存储器隔离机制等）。某些基础组件可比其它组件更好地执行这一功能，因此在其上打开数据对象的任一机器可作出关于是否相信该对象实际上是依照应用到该对象的安全需求来创建的判定。例如，如果对象包括通过键盘输入的文本，则高保证环境将以安全的方式从键盘接收输入，但是该环境安全地接收输入的能力可取决于基础组件保护从键盘到高保证环境的路径的能力。由于包装包含特定的基础组件生成的封印或签名，因此包装支持一种可信模型，其中，不同的机器可作出关于可靠数据对象如何基于在其上创建该数据对象的机器的假定安全性的判定。

下文描述支持分解或划分的应用程序的使用的系统、方法和机制。

示例性计算方案

图 1 示出了可在其中实现本发明的各方面的示例性计算环境。计算系统环境 100 仅为合适的计算环境的一个示例，并非建议对本发明的使用或功能的范围的局限。也不应将计算环境 100 解释为对示例性操作环境 100 中示出的任一组件或其组合具有依赖或需求。

本发明可以使用众多其它通用或专用计算系统环境或配置来操作。适合使用本发明的众所周知的计算系统、环境和/或配置包括但不限于：个人计算机、服务器计算机、手持式或膝上设备、多处理器系统、基于微处理器的系统、机顶盒、可编程消费者电子设备、网络 PC、小型机、大型机、包括任一上述系统或设备的分布式计算环境等等。

本发明可在计算机可执行指令的一般上下文环境中描述，计算机可执行指令如由计算机执行的程序模块。一般而言，程序模块包括例程、程序、对象、组件、数据结构等等，执行特定的任务或实现特定的抽象数据类型。本发明也可以在分布式计算环境中实践，其中，任务由通过通信网络或其它数据传输媒质连接的远程处理设备来执行。在分布式计算环境中，程序模块可以位于本地和远程计算机存储媒质中，包括存储器存储设备。

参考图 1，用于实现本发明的示例系统包括以计算机 110 形式的通用计算装置。计算机 110 的组件可包括但不限于，处理单元 120、系统存储器 130 以及将包括系统存储器的各类系统组件耦合至处理单元 120 的系统总线 121。处理单元 120 可表示如多线程处理器上支持的多个逻辑处理单元。系统总线 121 可以是若干种总线结构类型的任一种，包括存储器总线或存储器控制器、外围总线以及使用各类总线结构的局部总线。作为示例而非局限，这类结构包括工业标准体系结构（ISA）总线、微通道体系结构（MCA）总线、增强 ISA（EISA）总线、视频电子技术标准协会（VESA）局部总线以及外围部件互连（PCI）总线（也称为 Mezzanine 总线）。系统总线 121 也可以被实现为点对点连接、交换光纤等通信设备。

计算机 110 通常包括各种计算机可读媒质。计算机可读媒质可以是可由计算机 110 访问的任一可用媒质，包括易失和非易失媒质、可移动和不可移动媒质。作为示例而非局限，计算机可读媒质包括计算机存储媒质和通信媒质。计算机存储媒质包括以用于储存信息的任一方法或技术实现的易失和非易失，可移动和不可移动媒质，信息如计算机可读指令、数据结构、程序模块或其它数据。计算机存储媒质包括但不限于，RAM、ROM、EEPROM、闪存或其它存储器技术、CDROM、数字多功能盘（DVD）或其它光盘存储、磁盒、磁带、磁盘存储或其它磁存储设备、或可以用来储存所期望的信息并可由计算机 110 访问的任一其它媒质。通信媒质通常在诸如载波或其它传输机制的已调制数据信号中包含计算机可读指令、数据结构、程序模块或其它数据，并包括任一信息传送媒质。术语“已调制数据信号”指

以对信号中的信息进行编码的方式设置或改变其一个或多个特征的信号。作为示例而非局限，通信媒质包括有线媒质，如有线网络或直接连线连接，以及无线媒质，如声学、RF、红外和其它无线媒质。上述任一的组合也应当包括在计算机可读媒质的范围之内。

系统存储器 130 包括以易失和/或非易失存储器形式的计算机存储媒质，如只读存储器（ROM）131 和随机存取存储器（RAM）132。基本输入/输出系统 133（BIOS）包括如在启动时帮助在计算机 110 内的元件之间传输信息的基本例程，通常储存在 ROM 131 中。RAM 132 通常包含处理单元 120 立即可访问或者当前正在操作的数据和/或程序模块。作为示例而非局限，图 1 示出了操作系统 134、应用程序 135、其它程序模块 136 和程序数据 137。

计算机 110 也可包括其它可移动/不可移动、易失/非易失计算机存储媒质。仅作示例，图 1 示出了对不可移动、非易失磁媒质进行读写的硬盘驱动器 141、对可移动、非易失磁盘 152 进行读写的磁盘驱动器 151 以及对可移动、非易失光盘 156，如 CD ROM 或其它光媒质进行读写的光盘驱动器 155。可以在示例性操作环境中使用的其它可移动/不可移动、易失/非易失计算机存储媒质包括但不限于，磁带盒、闪存卡、数字多功能盘、数字视频带、固态 RAM、固态 ROM 等等。硬盘驱动器 141 通常通过不可移动存储器接口，如接口 140 连接到系统总线 121，磁盘驱动器 151 和光盘驱动器 155 通常通过可移动存储器接口，如接口 150 连接到系统总线 121。

图 1 讨论并示出的驱动器及其关联的计算机存储媒质为计算机 110 提供了计算机可读指令、数据结构、程序模块和其它数据的存储。例如，在图 1 中，示出硬盘驱动器 141 储存操作系统 144、应用程序 145、其它程序模块 146 和程序数据 147。注意，这些组件可以与操作系统 134、应用程序 135、其它程序模块 136 和程序数据 137 相同，也可以与它们不同。这里对操作系统 144、应用程序 145、其它程序模块 146 和程序数据 147 给予不同的标号来说明至少它们是不同的副本。用户可以通过输入设备，如键盘 162 和定位设备 161（通常指鼠标、跟踪球或触摸板）向计算机 110 输入命令和信息。其它输入设备（未示出）可包括麦克风、操纵杆、游戏垫、圆盘式卫星天线、扫描仪等等。这些和其它输入设备通常通过耦合至系统总线的用户输入接口 160 连接至处理单元 120，但是也可以通过其它接口和总线结构连接，如并行端口、游戏端口或通用串行总线（USB）。监视器 191 或其它类型的显

示设备也通过接口，如视频接口 190 连接至系统总线 121。除监视器之外，计算机也包括其它外围输出设备，如扬声器 197 和打印机 196，通过输出外围接口 195 连接。

计算机 110 可以在使用到一个或多个远程计算机，如远程计算机 180 的逻辑连接的网络化环境中操作。远程计算机 180 可以是个人计算机、服务器、路由器、网络 PC、对等设备或其它公用网络节点，并通常包括许多或所有上述与计算机 110 相关的元件，尽管在图 1 中仅示出了存储器存储设备 181。图 1 描述的逻辑连接包括局域网（LAN）171 和广域网（WAN）173，但也可包括其它网络。这类网络环境常见于办公室、企业范围计算机网络、内联网以及因特网。

当在 LAN 网络环境中使用时，计算机 110 通过网络接口或适配器 170 连接至 LAN 171。当在 WAN 网络环境中使用时，计算机 110 通常包括调制解调器 172 或其它装置，用于通过 WAN 173，如因特网建立通信。调制解调器 172 可以是内置或外置的，通过用户输入接口 160 或其它合适的机制连接至系统总线 121。在网络化环境中，描述的与计算机 110 相关的程序模块或其部分可储存在远程存储器存储设备中。作为示例而非局限，图 1 示出了远程应用程序 185 驻留在存储器设备 181 中。可以理解，示出的网络连接是示例性的，也可以使用在计算机之间建立通信链路的其它装置。

经划分或经“分解”的应用程序

软件，如应用程序，通常执行各种不同的功能并操作各种不同类型的数据。在这一意义上，应用程序可被视为不同功能的集合。将一个应用程序分解成其各种不同的功能是有用的，使得这些不同的功能可被单独执行（如，分配给提供不同安全级别的环境）。图 2 示出了一个应用程序如何被分解成其各种不同的功能。

应用程序 135 包括各种不同的功能 202(1)、202(2)、…、202(n)、202(n+1)、202(n+2)、…、202(n+m)。例如，应用程序 135 可以是文字处理程序，单独的功能可以是编辑、察看、打印、跟踪变化等等。应当注意，尽管图 2 示出应用程序 135 具有 n+m 个离散的功能，在判定什么是离散功能中应当谨慎判断。例如，所有的打印功能可被视为单个功能，或者打印可被视为包括两个或多个不同的功能（如，打印一页或者打印整个文档）。另外，如后文进一步描述的，同一基本操作可被视为多个功能，取决于正在操作什么类型的数据。（如，察看文档的基本操作可被认

为是两个不同功能之一，取决于被察看的数据是安全还是非安全的。）由此，可以有两个单独的察看功能，一个在某一方面是安全的，一个不是。本质上，将一个程序分解成其组成功能是围绕程序所做的事情（和/或程序所操作的各种不同类型的数据）画出边界的问题，并且一般没有关于如何画这些边界的特定要求。

在一个示例中，功能可分组在一起，使得功能202(1)到202(n)是第一分区206(1)的一部分，功能 202(n+1)到功能 202(n+m)是第二分区 206(2)的部分。可以方便地以这一方式分组功能，使得可以类似地处理同一分区中的功能。例如，分区 206(1)可包括应用程序 135 的涉及普通、非安全数据的功能，而分区 206(2)可包括应用程序 135 的涉及机密或安全数据（或需要其它某一级别的保护的数据）的功能。由此，分区 206(1)中的功能可在普通开放环境（如，由普通商业操作系统提供的环境）中执行，而分区 206(2)中的功能可在高保证环境中运行。（高保证环境在下文更具体讨论。）

图 3 示出了如何使用分区来允许单个应用程序 135 处理安全和非安全数据的示例。在图 3 的示例中，应用程序 135 执行需要高度保护的操作 302（如，涉及机密数据的操作），并且也执行需要低程度保护或不需要保护的操作 304（如，不涉及机密数据的操作）。（在本示例中，数据上的“操作”可包括任一类型的数据处理，如执行数据的输入或输出，执行数据的计算等）。在本示例中，不涉及机密数据的操作的功能由应用程序 135 的分区 206(1)处理，涉及机密数据（或需要某一类型的保护的数据）的操作的功能由分区 206(2)处理。例如，如果应用程序 135 是文字处理程序，则显示普通（非保护）文档可由分区 206(1)处理，而显示机密文档可由分区 206(2)处理。作为另一示例，如果应用程序 135 是股票贸易程序，则分区 206(1)可包含允许用户查寻股票的当前价格的功能，而分区 206(2)可在验证用户之后允许用户购买或出售股份（在这一情况下，用户的验证凭证以及他或她的财物账号是机密数据的一种类型）。

应当理解，图 3 示出了应用程序的不同分区可用于处理机密和非机密数据，但是图 3 不限于用于实现或使用分区应用程序的任一具体机制。下文结合图 4-8 讨论了一种以提供集成用户体验的方式支持应用程序的划分和分区的使用的示例体系结构。

示例性经划分的应用程序

应用程序的各种功能可依照任一标准来划分。在图 3 的示例中，依照该应用程序是处理安全还是非安全数据来划分应用程序的功能。当划分应用程序时，期望将各种划分集成到单个用户体验中。图 4 示出了经划分的应用程序的一个示例用户界面。（应当理解，“安全”数据指需要某一类型的保护的数据，尽管本发明不限于任一具体类型的保护。例如，在“保密”的意义上，数据可需要保护，在这一情况下，可加密该数据。作为另一示例，在可核实的意义上数据可需要保护－即，确定该数据自从时间中的某一参考点以来未被修改的能力－在这一情况下，可签署该数据。）

用户界面 400 是文字处理程序的界面。（应当理解，文字处理程序是应用程序的一个方便的示例，尽管它不意味着唯一的示例。）在图 4 的示例中，用户界面 400 显示了作为文字处理文档的一部分的各种项目 402(1)到 402(5)。项目 402(1)是普通的非安全文本。项目 402(2)是非安全图形 404。项目 402(3)、402(4)和 402(5)是安全文本的数据。应用程序的非安全分区执行不涉及安全数据的处理的所有处理。在本示例中，该处理包括非安全项目 402(1)和 402(2)的显示（并可能有编辑、打印、保存等），以及所有项目（甚至是安全项目）的布局。非安全分区能够展示安全项目，因为假定尽管项目 402(3)、402(4)和 402(5)的内容是机密的，这些项目的存在不是机密的。由此，非安全分区能够将项目 402(3)到 402(5)显示为弯曲线条 405，由此向用户提供了关于非安全分区不能处理的内容的至少某些用户体验。

如果用户希望察看内容项目 402(3)、402(4)和 402(5)，例如，用户可点击表示给定内容项目的弯曲线条 405。该行动可促使调用安全分区。安全项目之一中包含的内容然后可被传递到安全分区用于处理。在一个示例中，安全分区然后可在叠加在由非安全分区展示的内容的位置上的窗口中显示实际内容项目－如，当安全分区显示内容项目 402(3)时，它可通过在先前显示表示内容项目 402(3)的弯曲线条 405 的区域的上方放置该内容项目的图像。尽管上述情形涉及用户与软件的两个不同片断（即，安全和非安全分区）的交互动作，对用户而言可似乎他与单个应用程序交互，由此提供了跨两个分区的集成用户体验。

图 4 示出了经划分的应用程序为文字处理程序的示例。其它示例包括：

电子表格，其中用户可在非安全分区中输入公式并构建表格等，但是操作的实际数据仅对安全分区可用。由此，非安全分区在每一单元中显示某一占位符数据（如，“xxxx”）。单元中数据的估算和呈现由应用程序的安全分区执行。当用户

按“计算”按钮时，弹出一个窗口（由安全分区生成），并显示适当的行和列中的单元值，它们基于对安全分区可用的基础数据来计算。在这一实现中，安全分区将具有一估算引擎以及一（平凡的）呈现例程，但是不需要结合电子表格用户界面的其它方面、帮助窗口、公式构造器等。

安全贸易应用程序，其中，相应地向安全和非安全分区分配功能。例如，显示贸易图表和股票信息（公共可用）可由非安全分区执行。另一方面，安全分区允许用户输入股票贸易符号、用户要购买或售出的股票份额的价格和数量，并在第一次核实服务器的身份之后向远程贸易服务器发送该信息。

以图像格式（如，可移植文档格式或“PDF”）输出文档的文字处理器。安全分区能够读并显示该图像。这一文档的呈现被称为文档的“传真”。基础文字处理文档（即，包含格式化代码和文本字符而非仅图像的文档的版本）机器传真由非安全分区发送到安全分区，并且安全分区显示该传真。用户使用安全环境中的安全签署代理签署该传真（或文档和传真的摘要）。签署的传真（称为“传真-0”）和基础文档以大型二进制对象（blob）返回到非安全分区。当核实器接收包含该文档、传真-0 和传真的签名的大型二进制对象时，核实器首先基于基础文字处理文档呈现该传真的一个新副本（称为传真-1）。核实器将传真-0、传真-1、Word 文档和签名发送到安全分区，它核实传真是相同的，并且传真-0 上的签名是有效的。尽管该模型不提供文档的保密（由于文档的传真-1 仍可由非安全分区呈现），它的确服务来核实文档的完整性一即，文档是原始创建的那一个，并且未被修改的事实。

图 5 示出了支持应用程序的划分的体系结构。在图 5 中，有可在其中运行软件的两个环境 502(1)和 502(2)。在本示例中，有四个单独的处理器 504(1)、504(2)、504(3)和 504(4)。在本上下文中，“处理器”不仅指微处理器，还指以某一方式为应用程序处理数据的软件组件。处理器 504(1)和 504(2)运行在环境 502(1)中，处理器 504(3)和 504(4)运行在环境 502(2)中。例如，处理器 504(1)和 504(3)可分别为单个文字处理应用程序的非安全和安全部分。基础组件 508 主管环境 502(1)和 502(2)，使得两个环境可在单个机器上共存。本发明不限于任一具体类型的基础组件 508；基础组件 508 的一些示例是虚拟机器监控器（VMM）、外内核、微内核或系统管理程序。例如，环境 502(1)和 502(2)可以是 VMM 或系统管理程序主管的单独的操作系统。作为另一示例，基础组件 508 可以是操作系统之一一如，提供某些用户功能，并也实施其本身和其它（低保证）操作系统之间的分隔的高保证操作系统。

作为基础组件 508 的部分运行的一个组件是基准监控器 510。基准监控器 510 执行将给定的数据对象路由到正确的环境用于处理的功能。例如，在运行应用程序的过程中可遇到（或生成，或输入）标签数据 506，基准监控器 510 促使标签数据 506 被路由到正确的环境。标签数据 506 与允许基准监控器 510 确定应当将该标签数据路由到哪一环境的标识标记关联。另外，标签数据 506 也可包含允许目标环境确定应当将该数据给予哪一处理器来处理的额外的标记。标签数据 506 的示例结构在下文结合图 7 讨论。

标签数据 506 可在任何位置遇到（或生成，或输入），尽管最典型的标签数据将在一个环境中遇到，并被传递到另一环境。例如，文字处理应用程序的非安全部分可遇到包含某一机密文本的标签数据。文字处理应用程序然后可以图 4 所示的方式显示该文本的表示（如无法破译的弯曲线条）。万一用户期望显示实际文本（如，通过点击为该数据保留的区域），则应用程序的非安全分区（如，运行在环境 502(1)中的处理器 504(1)）可向基准监控器 510 提供标签数据 506，它然后向环境 502(2)提供该数据。环境 502(2)然后可将该数据路由到正确的处理器（如，处理器 504(3)），它然后能够以上文结合图 4 描述的方式在屏幕的适当位置上显示该数据。

应当理解，在图 5 的模型内，应用程序本质上由处理器构成－即，可处理不同的指定类型的数据或执行某些类别的任务的组件－并且应用程序的不同分区使用不同的处理器。例如，处理器 504(1)和 504(3)可表示单个应用程序（如，文字处理应用程序）的安全和非安全处理器，其中，根据数据是安全还是非安全数据将数据路由到处理器 504(1)或 504(3)的任一个。

也应当理解，没有关于什么类型的环境可用的要求，但是一个环境是高保证操作系统，另一环境是普通的全服务开放操作系统是尤其有用的。“高保证”操作系统是提供将正确执行其期望功能的相对高级的保证的系统。如果所有的软件（包括操作系统）与描述该软件的期望功能的规范相关联，并且如果所有软件遭受导致软件以非预期的方式表现的某一级别的程序错误或攻击，则“高保证”操作系统是提供操作系统将依照其规范表现的相对高级可信度的系统。（大多数商业软件具有显式、书面规范，尽管本上下文中的规范也可包括关于软件应当如何表现的隐式、非书面理解。）高保证与安全性不同，尽管高保证可用于实现安全性。例如，用于机密数据的处理器可被设计成运行在高保证操作系统中；在处理器保护机密数据的能力取决于它所运行的环境的正确行为（如，系统调用的正确执行、正确的进程隔

离等）的方面，处理器可提供如果处理器运行在普通、全服务操作系统中就无法实现的一种安全级别。（高保证的折衷选择使高保证环境具有十分有限的功能范围；较大的程序是最难核实该程序将在各种各样的情况下正确表现。由此，诸如WINDOWS XP 等全服务商业操作系统通常被认为不是高保证的。）

在以上高保证的描述的上下文中，可以理解，将一个应用程序划分成处理不需要重大安全性的数据的低安全性处理器，以及需要更多安全性的数据的高安全性处理器通常是有用的。高安全性处理器可运行在高保证环境中，低安全性处理器可运行在低保证环境中。

支持经划分的应用程序的示例体系结构

图 6 示出了可执行划分应用程序的示例体系结构 600。体系结构 600 包括基础组件 508，其功能是主管可执行应用程序的划分的各种环境。如上所述，基础组件 508 可采用各种形式，如 VMM、外内核、微内核、系统管理程序等等。基础组件 508 具有主管多个环境（如，操作系统）的功能，并也管理（并限制）这些环境之间的交互。多个环境的主管可使用各种技术来执行。例如，基础组件 508 可向多个操作系统的每一个展现自含“虚拟机器”；操作系统然后控制虚拟机器“虚拟硬件”，并且基础组件 508 向“真实的”硬件发出基于（但不必要与其相同）操作系统给予虚拟机器的指令的指令。（一般而言，这就是传统的 VMM 如何工作的。）作为另一示例，基础组件 508 可向不同的操作系统分配某些设备以及机器的物理地址空间的某些片段，并可通过准许每一操作系统仅控制其分配的设备和其分配的地址空间部分来实施这一分配。本发明不限于基础组件的任一具体实施例；为图 6 的目的，仅假定有一能够主管多个环境的基础组件，使得这些多个环境可以彼此某种程度的隔离在单个机器上共存。

在图 6 的示例中，基础组件 508 主管操作系统 602(1)到 602(4)。操作系统 602(1)是 WINDOWS XP 操作系统或另一通用操作系统的一个实例；操作系统 602(2)是 Linux 操作系统的一个实例；操作系统 602(3)是“网络点”一即，提供有限功能但是将正确实现该功能的高保证的高保证操作系统（在上述意义内）；操作系统 602(4)可以是另一通用操作系统，如 OS/2。一般而言，基础组件 508 可主管任意数量的操作系统（或其它类型的环境），并且图 6 所示的四个操作系统仅为示例。

在给定的操作系统内，可能运行应用程序级软件的片段。例如，MICROSOFT

WORD 文字处理程序（604）和 MICROSOFT EXCEL 电子表格程序（606）在 WINDOWS XP 操作系统 602(1)下运行。Linux 操作系统 602(2)、网络点 602(3)和 OS/2 操作系统 602(4)也可运行其自己的应用程序。在本示例中，称为“Word-let”的程序（608）和“Excel-let”（610）在网络点 602(3)下运行。Word-let 608 是当 WORD 需要处理安全程序时与 WORD 604 一起工作的安全程序。类似地，Excel-let 610 为 EXCEL 606 处理安全数据。由此，在图 4 的示例中，WORD 604 可以是处理非安全数据项目、在窗口内展示数据项目并显式表示安全项目的不可破译的弯曲线条的程序（或“处理器”）。Word-let 608 可以是打开安全项目并在 WORD 604 为该项目保留的窗口的区域中呈现该项目的程序。本质上，WORD 604 和 Word-let 608 一起表示跨在不同的环境中执行的不同处理器的文字处理应用程序的功能的划分（或“分解”）。类似地，EXCEL 606 和 Excel-let 610 可具有类似的关系－即，EXCEL 606 处理不涉及安全数据的大多数电子表格功能，而 Excel-let 610 代表 EXCEL 处理安全数据。

如上所述，网络点 602(3)是可提供可执行可信应用程序（即，对于诸如打开机密数据等敏感操作可足够确信其行为可信的应用程序）的环境的高保证操作系统。然而，网络点 602(3)可能提供十分有限的功能。由此，例如，将文字处理应用程序划分成 WORD 604 和 Word-let 608 是有用的，使得 WORD 可使用通用操作系统中可用的广泛特征来提供广泛的功能，并且 Word-let 可使用网络点 602(3)所提供的环境的高保证特性来执行（可能有限的）具有高度可信性的敏感功能。

图 6 示出了提供支持应用程序的划分的基础结构的基础组件。以下是该基础结构可包括的一些特性的描述：

注册和目录服务：基础组件可提供一种接口，基础组件所主管的环境可通过该接口向基础组件注册。基础组件也可提供用以启动主管的环境的方法（或者基础组件可以是主管环境之一）。环境的保证级别是与该环境关联的元信息，并且基础组件可提供可任选的服务来以结构化方式访问并查找该信息。

分隔：主管环境具有基础组件向其分配的安全上下文。环境的安全上下文可包含以下组件：

a．DAC 上下文，如环境的代码 id。

b．MAC 上下文：在 MAC 的情况下，环境的 MAC 上下文包括敏感标签和类别。

通信：环境间通信由基础组件通过使用它所拥有的单向传输来控制。

a．传输提供了环境之间的顺序且可靠消息传送。提供同步对象用于向环境通知关于传输上数据的到达。

b．访问控制：传输是基础组件所拥有的对象，并且主管环境对传输的读取和写入的能力由基础组件所实施的访问控制模型来控制。在 DAC 的情况下，这由传输上的 ACL 控制用于行动“读”和“写”。在 MAC 的情况下，这由附加到传输的标签控制。为提供对多级设备的输出，基础组件提供前端安全过滤器的实现。前端安全过滤器（FESF）是可被附加到传输的读或写端的向基础组件注册的功能。在写（读）-FESF 的情况下，写 FESF 由 VMM 在数据由 VM 写入（读取）传输之前调用。写-FESF 的一个示例是 *Seal()*，它通常在如果 MAC 有效时在数据写入普通（非高保证）环境之前应用到高保证环境中的数据。如果 MAC 未有效，则环境必须在向另一环境发送之前加密它认为私密敏感的数据。

消息：传输跨不同的环境携带消息。消息是由基础组件创建的数据大型二进制对象（data blob），它被结构化如下：

a．由基础组件分配的安全上下文：包含 DAC 和 MAC 上下文。DAC 上下文包含创建环境的主题 id。MAC 上下文包含创建环境的敏感标签和类别。

b．内容：数据（包括由创建环境分配的安全上下文）。

更高级抽象：环境（或运行在环境中的应用程序或其它软件对象）可通过使用基础组件展现的通信抽象由其自己实现 RPC 接口。这可以是环境所提供的服务代码，或仅为库代码。可展现的另一抽象是环境之间的套接字调用接口，它不需要由基础组件实现。

焦点管理：在经划分的应用程序的许多示例中，从一个环境到另一个的消息的到达可促使安全输入或输出设备的当前所有者中的变化。焦点管理由基础组件在环境的请求之后提供－如，环境可在来自另一环境的消息到达后请求焦点中的变化。另一环境然后具有与该安全输入/输出设备的会话，并放弃控制。基础组件可强迫终止环境的 I/O 会话。

用于经划分的应用程序的示例数据对象

图 7 示出了数据对象的示例结构，它允许该数据对象用于经划分的应用程序。如上所述，当应用程序被划分时，应用程序的第一分区可遇到不能由该分区处理但

需要被发送到第二分区的数据对象。在第一分区是应用程序的“非安全”部分，而第二分区是“安全”部分的情况下，第一分区较佳地可认识到数据对象需要被发送到别处用于处理，即使第一分区无法确定关于该对象的其它情况（如，即使第一分区无法读该对象的内容）。另外，在某些情况下，重要的是核实该数据对象实际上在安全情况下创建，并未被修改（如，应当可确定数据是否输入到一种保护在从键盘到应用程序的 I/O 流的路径上的数据免遭欺骗的环境）。由此，同样较佳地，数据对象以允许核实其处理链的方式显示。

数据对象 700 包括数据项目 702。数据项目 702 是数据对象 700 为携带目的而存在的基础实质内容一如，机密文字处理文档（或其部分）、电子表格的敏感财政数据、金融交易中验证用户的密码信息等等。

数据项目 702 由处理该数据的体系结构的每一层在一系列包装中包装。每一包装用于两个目的：（1）包装标识在该项目上放置该包装的实体（如，应用程序、环境等），它随后协助将项目路由到该实体；（2）包装以随后允许核实该项目自从被包装以来未被修改的方式密封该项目。例如，如果 Word-let 608 创建数据项目 702，则 Word-let 可附加数据项目 702 的数字签名以及将 Word-let 标识为处理器的头，随后当需要以某一方式处理数据项目 702 时，它应当被路由到该处理器。由此，该签名和头构成了包装 704。应当注意，包装不限于签名和头，或任一特定的实施例，有各种允许标识项目并允许核实项目的完整性（如，未修改）的已知技术。

如上所述，诸如 Word-let 等处理器的安全性基于可信组件的层次结构一即，Word-let 是可信的，因为它依赖于网络点的高保证；网络点是可信的，因为它依赖于基础组件的隔离能力，等等。由此，链的层次结构中的每一组件逐渐通向创建数据项目 702 并以其自己的包装来包装该项目的应用程序。已由 Word-let 的包装来包装的数据项目 702 然后由网络点的包装 706 来包装。该（双重包装的）项目然后由基础组件的包装 708 来包装。每一包装本质上标识了包装组件，并也构成包装的内容自从它们被包装以来未改变的可核实断言（在包装组件是可信的意义上）。

另外，由于包装是依照层次的顺序嵌套的，每一包装可用于路由目的。由此，当应用程序遇到数据对象 700 时，外部包装 708 将该对象标识为需要被路由到另一处理器的对象，因此该对象被发送到基础组件。基础组件然后使用包装来核实包装内的内容的完整性，并使用下一层包装 706 来确定该内容需要被路由到哪一环境（在本情况下为到网络点）。网络点然后核实包装 706 内的内容的完整性，并也使

用包装 704 来确定哪一软件对象（在本情况下为 Word-let）将处理内容。Word-let 然后核实包装 704 内的内容的完整性。该内容是数据项目 702。由此，数据对象 700 的结构允许数据对象 702 在逐渐通向处理数据项目 702 的处理器的链的每一步骤中被路由并核实。

使用经划分的应用程序的示例过程

图 8 示出了可使用经划分的应用程序的一个示例过程。为图 8 的示例目的，假定应用程序具有两个处理器，称为处理器 1 和处理器 2。

最初，处理器 1 运行；在处理器运行的过程中，处理器 1 遇到数据对象（802）。如上所述，处理器 1 可能无法读数据对象，但是能够认识到数据对象为需要发送到别处以用于处理的某物。由此，处理器 1 将数据对象发送到基准监控器（804）。

基准监控器使用该数据对象的最外部包装来确定该数据对象需要被发送到何处。如上所述，基准监控器（较佳地为上述基础组件的一部分）也使用最外部的包装来核实该包装的内容的完整性。在本示例中，假定维持了该内容的完整性，并且基准监控器确定该数据对象需要由处理器 2 来处理（806）。（如上所述，基准监控器可能不直接知晓处理器 2 的存在，但是可使用包装以将该对象路由到特定的环境，其中，该对象实际上由接收环境路由到处理器 2。由于上文结合图 7 讨论了以这一方式的嵌套包装的使用，为简化图 8 的示例的目的，仅假定数据方式可以某种方式路由到处理器 2，而不考虑将对象路由到处理器 2 时所涉及的中间路由步骤。）

当确定数据对象以处理器 2 为目的时，确定（808）处理器 2 是否正在运行。例如，该确定可由预期运行处理器 2 的环境作出（如，在当前示例中为网络点）。如果处理器 2 未运行，则启动处理器 2（810）。在启动了处理器 2 之后（或确定已运行），如果处理器 2 要处理任何输入或输出，则给予处理器 2 焦点（即，在输入和输出设备上执行 I/O 的能力）。

注意，上述示例仅为解释目的提供，并且不应当被解释为限制本发明。尽管参考各种实施例描述的本发明，可以理解，此处所使用的词语是描述和说明的词语，而非限制的词语。此外，尽管此处参考特定的装置、材料和环境描述了本发明，本发明并不意味着局限在此处所揭示的细节上；相反，本发明延及处于所附权利要求书的范围之内的所有功能等效结构、方法和使用。从本说明书的教导获益的本领域的技术人员可在不脱离本发明的各方面的范围和精神的情况下对其作出许多修改

和变化。

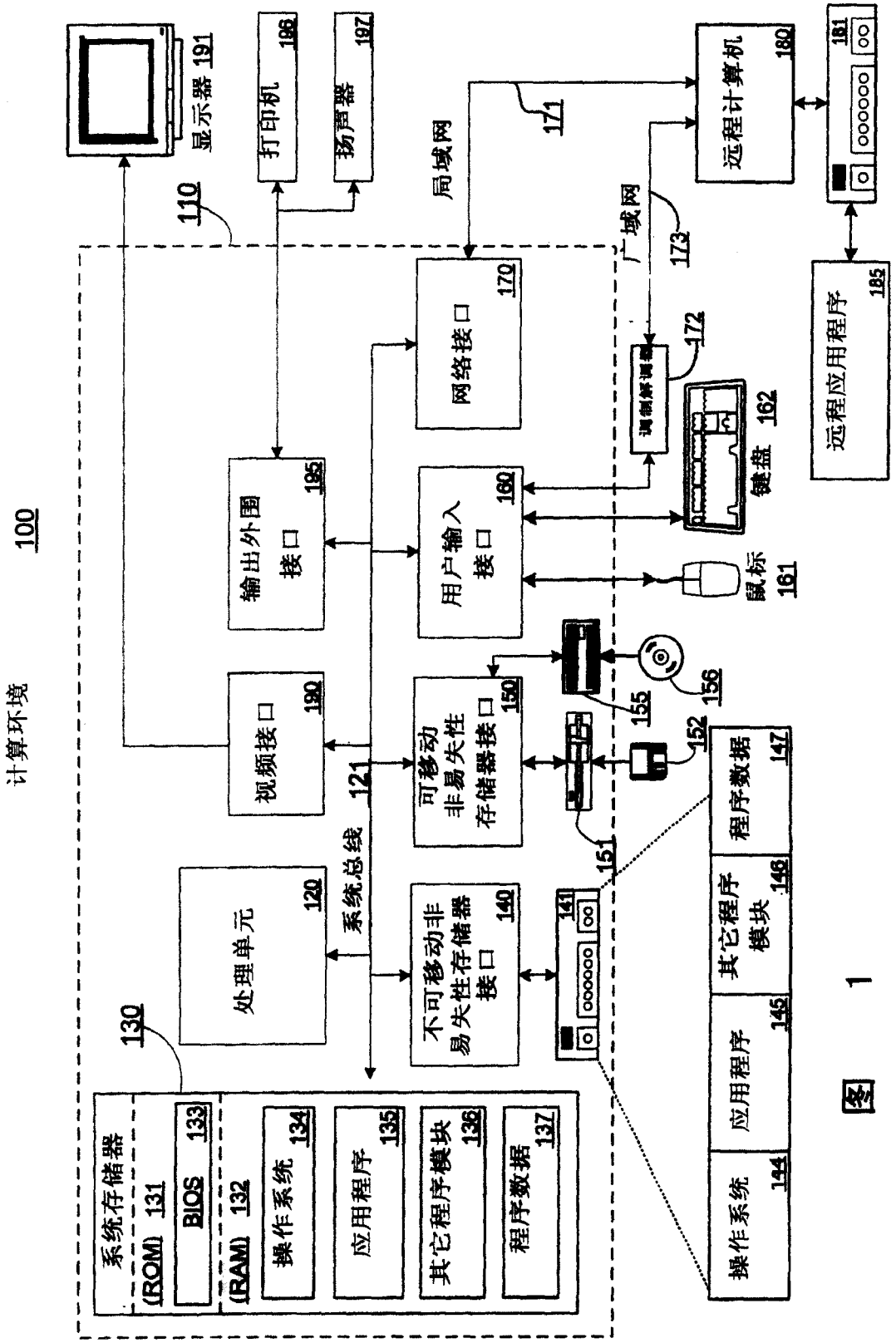

计算环境 100
110
显示器 191
打印机 196
扬声器 197
局域网 171
广域网 173
远程计算机 180
远程应用程序 185
网络接口 170
输出外围接口 195
用户输入接口 160
视频接口 190
可移动非易失性存储器接口 150
不可移动非易失性存储器接口 140
处理单元 120
系统总线 121
键盘 162
鼠标 161
系统存储器 130
(ROM) 131
BIOS 133
(RAM) 132
操作系统 134
应用程序 135
其它程序模块 136
程序数据 137
操作系统 144
应用程序 145
其它程序模块 146
程序数据 147

图 1

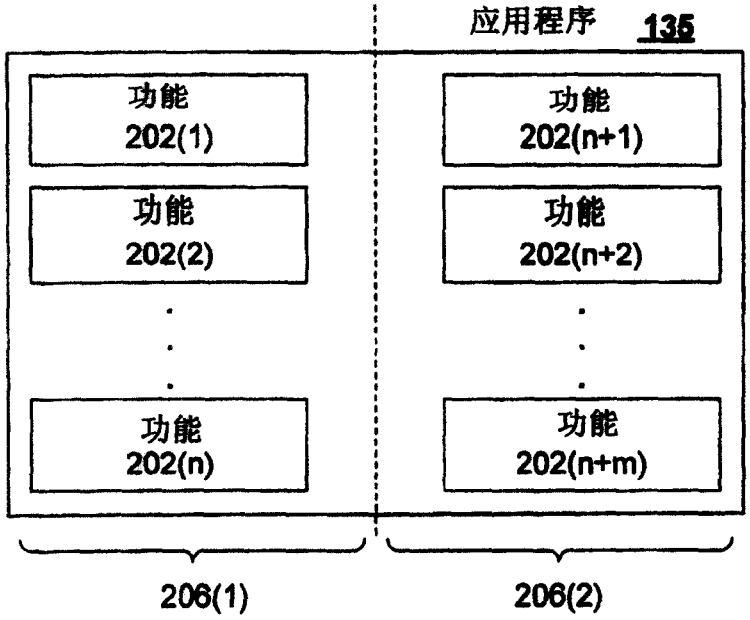

应用程序 135
功能 202(1)
功能 202(2)
功能 202(n)
功能 202(n+1)
功能 202(n+2)
功能 202(n+m)
206(1)
206(2)

图 2

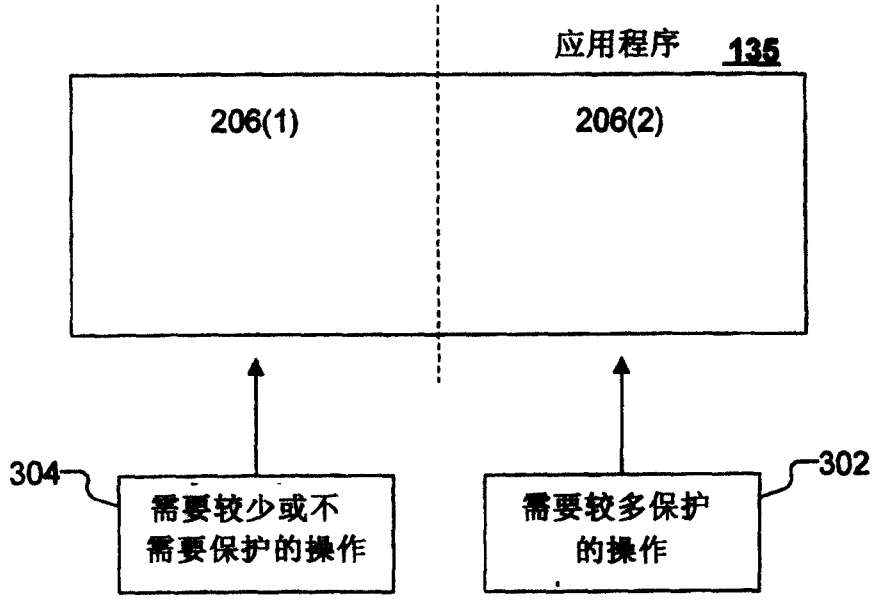

应用程序 135
206(1)
206(2)
304
需要较少或不需要保护的操作
302
需要较多保护的操作

图 3

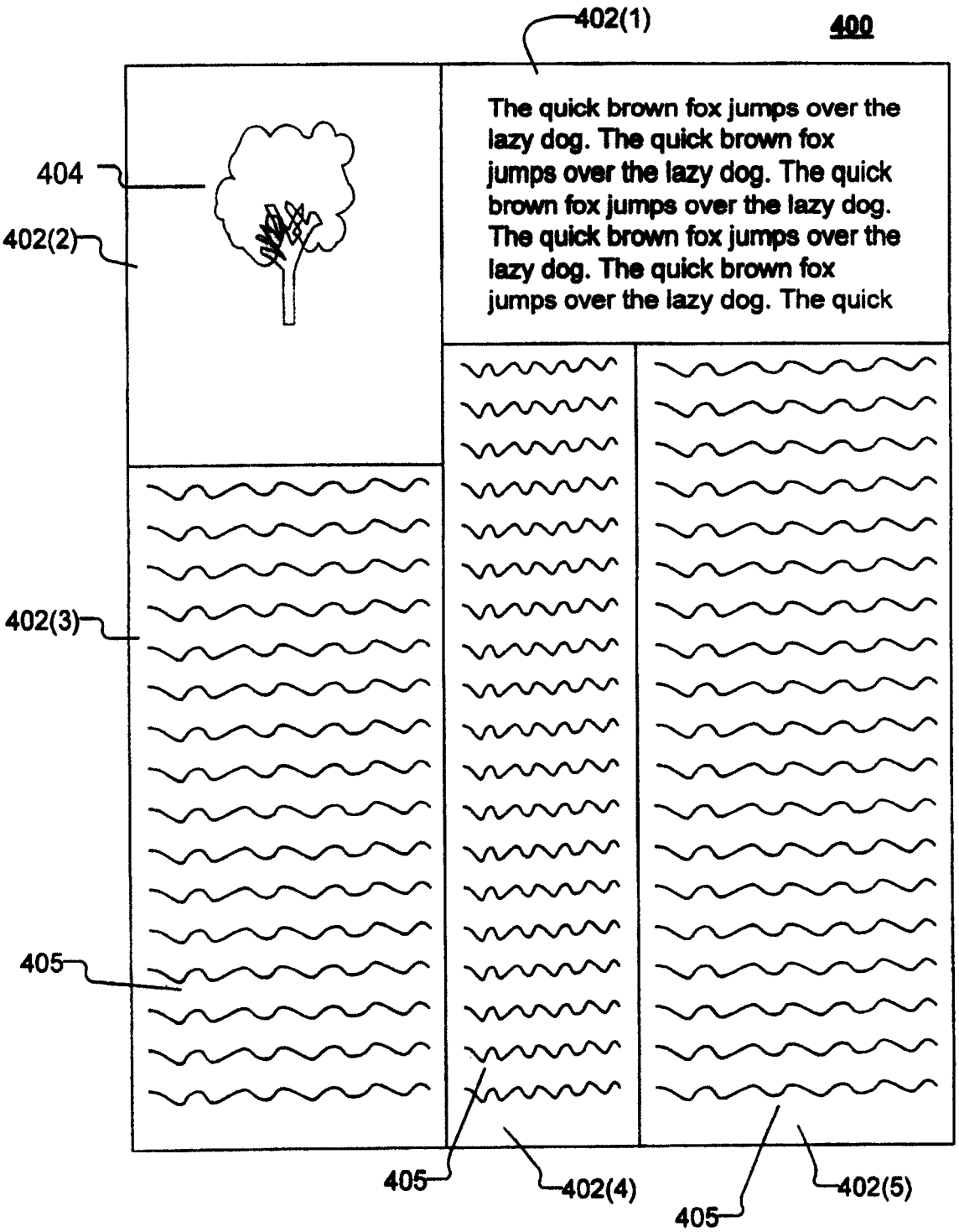

402(1)
400
The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick
404
402(2)
402(3)
405
405
402(4)
405
402(5)

图 4

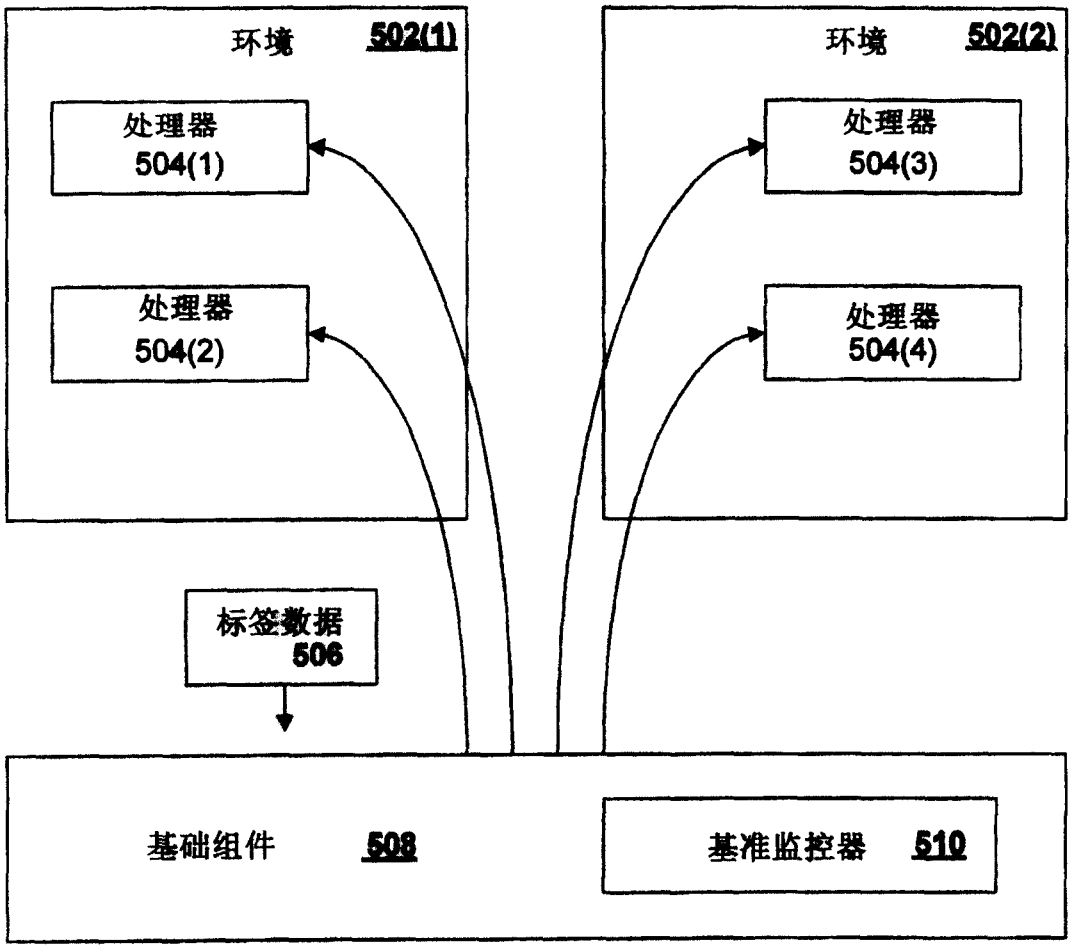

环境
502(1)
处理器
504(1)
处理器
504(2)
环境
502(2)
处理器
504(3)
处理器
504(4)
标签数据
506
基础组件
508
基准监控器
510

图 5

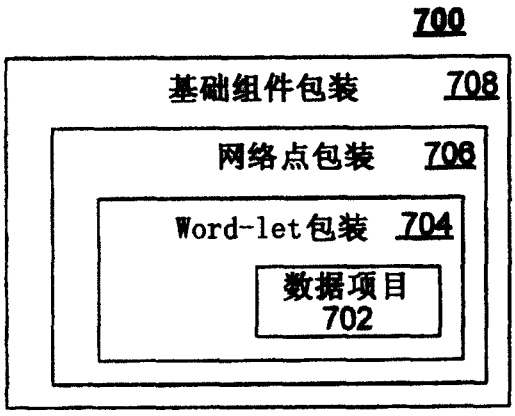

700
基础组件包装
708
网络点包装
706
Word-let包装
704
数据项目
702

图 7

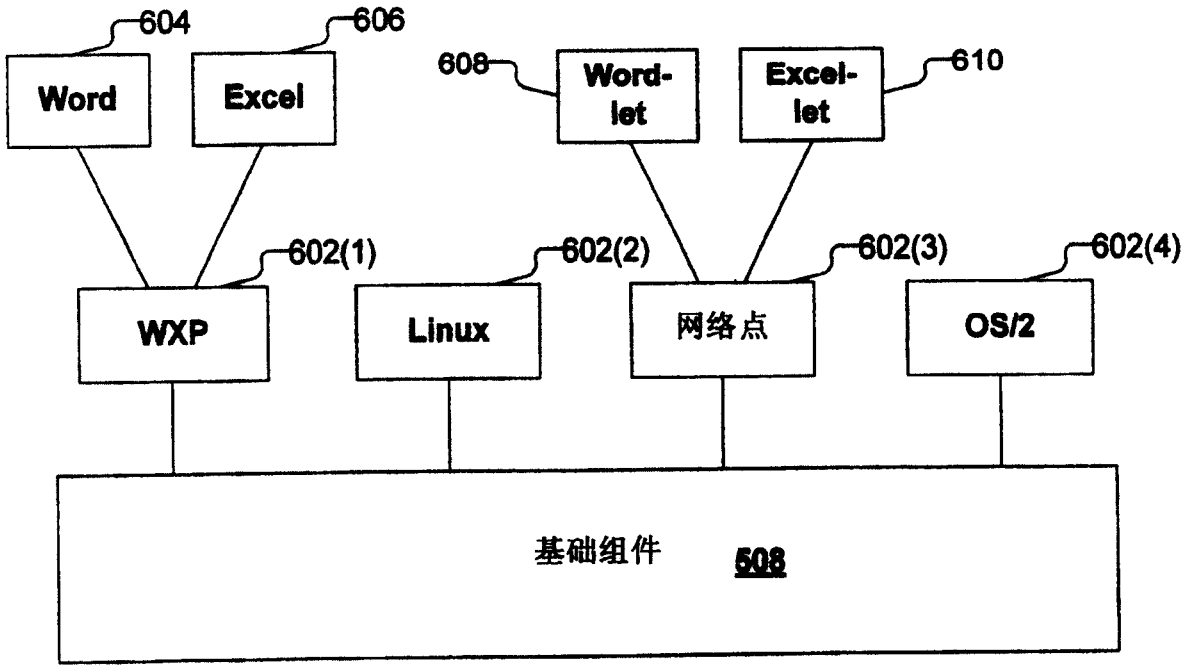

604
606
608
610
Word
Excel
Word-let
Excel-let
602(1)
602(2)
602(3)
602(4)
WXP
Linux
网络点
OS/2
基础组件
508

图 6

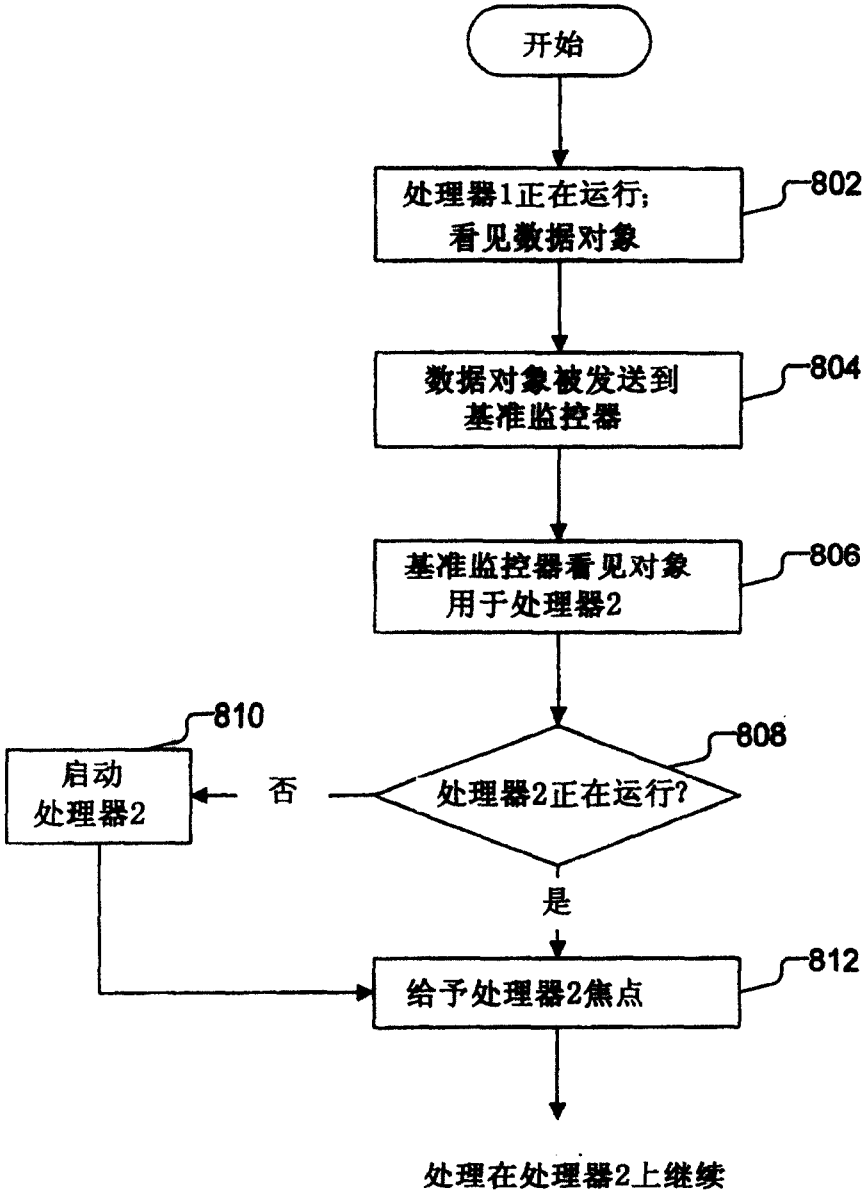

开始
处理器1正在运行;
看见数据对象
802
数据对象被发送到
基准监控器
804
基准监控器看见对象
用于处理器2
806
处理器2正在运行?
808
否
启动
处理器2
810
是
给予处理器2焦点
812
处理在处理器2上继续

图 8