



(12) 发明专利

(10) 授权公告号 CN 1612112 B

(45) 授权公告日 2011.09.28

(21) 申请号 200410083108.8

审查员 马美红

(22) 申请日 2004.09.28

(30) 优先权数据

10/697,197 2003.10.30 US

(73) 专利权人 微软公司

地址 美国华盛顿州

(72) 发明人 E·S·科恩

(74) 专利代理机构 上海专利商标事务所有限公
司 31100

代理人 谢喜堂

(51) Int. Cl.

G06F 12/08 (2006.01)

(56) 对比文件

US 6105103 A, 2000.08.15, 全文.

US 20030200402 A1, 2003.10.23, 全文.

WO 9743714 A1, 1997.11.20, 全文.

US 5404478 A, 1995.04.04, 全文.

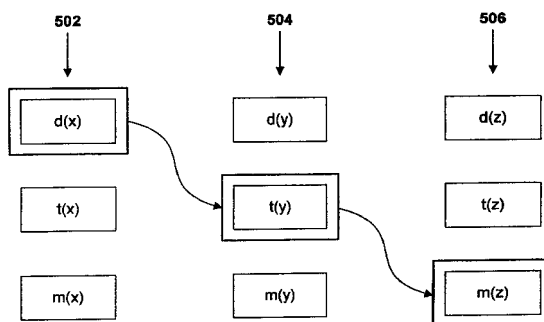
权利要求书 4 页 说明书 11 页 附图 5 页

(54) 发明名称

用于地址翻译控制的阴影页表

(57) 摘要

在采用虚拟地址的计算机系统中, 储存了给定页的多个版本: 目录版本、表版本和数据版本。所述数据版本包含软件对象认为储存在页中的数据。所述页的目录和表版本包含以符合虚拟地址系统所采用的地址翻译映射上的限制的某一方式修改的页内容的版本。当由虚拟地址系统将页用作目录或表时, 则分别使用该页的目录或表版本。当页是读请求的目标时, 使用该页的数据版本。



1. 一种为地址翻译映射创建阴影页的方法,所述地址翻译映射包括一页目录和多个页表,所述页目录包括到所述多个页表的链接,每一所述页表包括到多个数据页的链接,所述页目录和页表的每一个被储存在所述数据页之一中,其特征在于,所述方法包括:

对于所述多个页表的至少一个,基于所述多个页表中的该页表创建第一阴影页表,所述第一阴影页表在以下至少一个方面有别于所述多个页表中的该页表:

所述第一阴影页表中的至少一个条目链接到与在所述多个页表中的该页表的该条目的对应链接不同的数据页;以及

所述第一阴影页表包含一个或多个只读链接,其在所述多个页表中的该页表中对应的链接是读/写的;以及

基于所述页目录创建一阴影页目录,所述页目录包括到所述多个页表的该页表的链接,所述阴影页目录包括到所述第一阴影页表的链接而非到所述多个页表的该页表的链接;

其中当软件的正确行为取决于存储器,该存储器包含了该软件认为它已写入该存储器的值,并且修改一请求以编辑所述地址翻译映射将导致存储器包含不同于软件认为它所写入的值的值时,所述数据页的一个数据版本被展现给所述软件,所述数据页的数据版本包括所述软件相信存储到所述数据页中的数据,并且当翻译地址时,参考所述阴影页目录。

2. 如权利要求1所述的方法,其中,一政策支配对存储器的访问,其中,基于应用到所述虚拟地址的所述地址翻译映射的对所述存储器的访问导致所示政策的违反,并且其中,基于所述阴影页目录和被应用到所述虚拟地址的所述第一阴影页表对所述存储器的访问不导致所述政策的违反。

3. 如权利要求1所述的方法,其特征在于,所述数据页的每一个储存在存储器的一具体帧中,其中,所述页目录储存在第一帧中,并且其中,所述方法还包括:

在不同于所述第一帧的第二帧处维护所述页目录的副本;以及
将所述阴影页目录储存在所述第一帧处。

4. 如权利要求1所述的方法,其特征在于,所述页目录包括到第一尺寸页的链接,所述第一尺寸页包括多个第二尺寸页,并且其中,所述方法还包括:

创建包括到所述多个第二尺寸页的链接的第二阴影页表,其中,所述阴影页目录包括到所述第二阴影页表的链接。

5. 一种管理存储器的使用的系统,其特征在于,它包括:

一包括可读和可写的多个可分别寻址组件的存储器,每一所述可分别寻址组件具有一与其关联的物理地址;

一定义在虚拟地址和所述可分别寻址组件的物理地址之间的映射的地址翻译数据结构;

一接收访问所述可分别寻址组件的第一个的请求的存储器管理器,所述请求基于虚拟地址标识所述可分别寻址组件的第一个,所述存储器管理器基于包括所述地址翻译数据结构的阴影表示的数据将所述虚拟地址翻译成所述可分别寻址组件的第一个的物理地址,其中当软件的正确行为取决于存储器,该存储器包含了该软件认为它已写入该存储器的值,并且修改一请求以编辑所述地址翻译映射将导致存储器包含不同于软件认为它所写入的值的值时,所述页的一个数据版本被展现给所述软件,所述页的数据版本包括所述软件相

信存储到所述页中的数据,并且当翻译地址时,参考所述阴影表示。

6. 如权利要求 5 所述的系统,其特征在于,所述存储器被组织成多个页,所述可分别寻址组件的第一个位于所述多个页之一内,其中,所述地址翻译数据结构包括:(1) 包含到所述多个页的链接的多个页表,以及(2) 一包含到所述多个页表的链接的页目录,并且其中,所述地址翻译数据结构的阴影表示在至少一个链接上不同于所述地址翻译数据结构。

7. 如权利要求 6 所述的系统,其特征在于,每一所述链接包含在所述页目录中,并且所述页表包含一个或多个属性,并且其中,所述阴影表示中的至少一个链接在至少一个属性上不同于所述地址翻译数据结构中的对应链接。

8. 如权利要求 6 所述的系统,其特征在于,所述页目录和所述页表的每一个储存在所述多个页之一中,每一所述页具有与其关联的一物理位置描述符,所述页目录和页表中的每一链接基于所述物理位置描述符标识所述页之一。

9. 如权利要求 8 所述的系统,其特征在于,所述阴影表示包括所述页目录或所述页表的至少一个的替换版本,并且其中,所述替换版本储存在具有与所述替换版本所基于的页不同的物理位置描述符的页处。

10. 如权利要求 5 所述的系统,其特征在于,一政策支配所述存储器的可访问性,其中,所述地址翻译数据结构使存储器遭受所述政策的违反,并且其中,所述系统还包括:

一存储器访问控制管理器,它基于所述地址翻译数据结构创建所述阴影表示,并确保如果用于基于虚拟地址访问所述存储器,所述阴影表示不导致所述政策的违反。

11. 如权利要求 10 所述的系统,其特征在于,所述政策将所述存储器的一部分定义为不可访问,并且其中,所述存储器访问控制管理器确保所述阴影表示不向所述存储器的该部分展现虚拟地址。

12. 如权利要求 10 所述的系统,其特征在于,所述政策将所述存储器的一部分定义为可读但不可写,并且其中,所述存储器访问控制管理器确保所述阴影表示包含将所述存储器的该部分标记为只读的一个或多个属性。

13. 如权利要求 10 所述的系统,其特征在于,所述存储器访问控制管理器确保所述阴影表示包含将所述存储器部分标记为只读的一个或多个属性,所述存储器这些部分储存:(1) 所述地址翻译数据结构;以及(2) 所述阴影表示中的至少之一。

14. 一种执行存储器访问请求的方法,其特征在于,它包括:

接收读或写存储器单元的请求,所述请求基于虚拟地址标识所述存储器单元;

基于定义虚拟地址和物理地址之间的关系的映射的表示访问所述存储器单元,所述映射储存在所述存储器的一个或多个页内,所述映射的表示包括基于所述一个或多个页的第一个的至少一个阴影页,所述映射包括至少一个方面,其如果用于基于所述虚拟地址访问所述存储器,则将导致存储器访问政策的违反,所述阴影页与所述一个或多个页的所述第一个在下述方式上不同:它使基于所述虚拟地址来利用所述映射的表示对所述存储器的访问不违反所述存储器访问政策;当软件的正确行为取决于存储器,该存储器包含了该软件认为它已写入该存储器的值,并且修改一请求以编辑所述地址翻译映射将导致存储器包含不同于软件认为它所写入的值的值时,所述页的一个数据版本被展现给所述软件,所述页的数据版本包括所述软件相信存储到所述页中的数据,并且当翻译地址时,参考阴影页目录;以及

执行所述访问请求中指定的读或写。

15. 如权利要求 14 所述的方法,其特征在于,所述存储器访问政策将所述存储器的一部分定义为不可访问,其中,所述映射向定义虚拟地址映射的所述存储器部分展现可写链接,并且其中,所述映射的表示不向定义虚拟地址映射的所述存储器部分展现可写链接。

16. 如权利要求 14 所述的方法,其特征在于,所述映射包括:(1) 包含到一组所述一个或多个页的链接的多个表,以及(2) 包含到所述多个表的链接的目录,所述至少一个阴影页包括一阴影目录,所述阴影目录与所述目录在至少一个方面中不同:所述阴影目录的至少一个链接指向阴影页表而非所述多个表之一。

17. 如权利要求 14 所述的方法,其特征在于,所述映射包括包含到一组所述一个或多个页的链接的多个表,其中,所述阴影页包括基于所述表之一的表示,并且其中,所述阴影页包含对存在于所述表的所述之一中的第一链接的表示,其中,所述第一链接是所述表的所述之一中的读/写链接,并且其中,所述阴影页与所述表的所述之一在下述方面不同:在所述阴影页的表示中所述第一链接被标记为只读。

18. 如权利要求 14 所述的方法,其特征在于,所述阴影页包括一目录,其中,所述存储器单元拥有包括多个第二尺寸页的第一尺寸页,其中,所述映射包括包含到所述第一尺寸页的链接的目录,其中,所述阴影页以所述目录为基础,并且其中,所述阴影页与所述目录在下述方面不同:所述阴影页包含到一表的链接而非到所述第一尺寸页的链接,其中,所述表包含到包括在所述第一尺寸页中的第二尺寸页的链接。

19. 一种具有表示一地址翻译映射的数据结构的计算机,所述地址翻译映射包括一页目录,所述目录包括到多个页表的链接,每一所述页表储存在该计算机的计算机可读媒质的一具体帧内,每一所述页表包括到所述计算机可读媒质的多个页的链接,其特征在于,所述计算机包括:

提供一基于所述多个页表的第一个的阴影页表的装置;

提供一基于所述页目录的阴影页目录的装置,所述页目录包括到所述多个页表的该页表的链接,所述阴影页目录包括到所述阴影页表的链接而非到所述多个页表的该页表的链接,所述页目录包括包含到所述多个页表的第一个的链接的第一条目,所述阴影页表包括对应于所述第一条目的第二条目,所述第二条目包含到所述阴影页表的链接而非到所述多个页表的第一个的链接,其中当软件的正确行为取决于存储器,该存储器包含了该软件认为它已写入该存储器的值,并且修改一请求以编辑所述地址翻译映射将导致存储器包含不同于软件认为它所写入的值的值时,所述页的一个数据版本被展现给所述软件,所述页的数据版本包括所述软件相信存储到所述页中的数据,并且当翻译地址时,参考所述阴影页目录。

20. 如权利要求 19 所述的计算机,其特征在于,所述多个页表的第一个储存在第一帧中,其中,所述阴影页表储存在第二帧中,并且其中,所述阴影页目录在下述方面不同于所述页目录:所述页目录中的链接包含所述第一帧的标识符,而所述阴影页目录中的对应链接包含所述第二帧的标识符。

21. 如权利要求 19 所述的计算机,其特征在于,所述多个页表的第一个包含到所述页的第一个的链接,其中,所述阴影页表包含到基于所述页的第一个的表示的链接而非到所述页的第一个的链接,基于所述页的第一个的所述表示储存在与所述页的第一个不同的帧

中。

22. 如权利要求 21 所述的计算机,其特征在于,所述多个页的第一个储存所述页目录或所述多个页表的第一个。

23. 如权利要求 22 所述的计算机,其特征在于,所述多个页表的第一个包含将所述多个页的第一个指定为可读和可写的链接,并且其中,所述阴影页表中对应的链接将所述多个页的第一个指定为只读。

24. 如权利要求 19 所述的计算机,其特征在于,所述页目录和所述多个页表的第一个包含至少一个特征,使得如果用于基于虚拟地址访问存储器,将导致存储器访问政策的违反,并且其中,所述阴影页目录和阴影页表包含数据,使得当基于所述虚拟地址通过所述阴影页目录和所述阴影页表访问所述存储器时将不导致所述存储器访问政策的违反。

用于地址翻译控制的阴影页表

技术领域

[0001] 本发明一般涉及计算机中的存储器管理领域,尤其涉及虚拟地址系统中地址翻译表的管理。

背景技术

[0002] 现代计算机系统通常提供某一类型的虚拟地址机制。如本领域已知的,与计算机系统关联的每一个别可访问存储器单元具有唯一地标识该存储器单元的物理地址。然而,在支持虚拟寻址的计算机系统中,可能向物理位置分配虚拟地址。虚拟地址系统使用翻译映射以将虚拟地址翻译成物理地址。

[0003] 虚拟寻址系统的一个特征是可能配置地址翻译映射使得某些物理地址组(如,物理存储器页)不具有虚拟地址。在典型的基于分页的存储器管理模式中,地址翻译映射将虚拟页描述翻译成物理页帧号。由此,通过确保该地址翻译映射不通向页帧,使给定物理页帧中的所有位置可以拒绝虚拟地址。更一般地,许多虚拟寻址模式用可通过虚拟地址执行的访问(如,读、读/写)来标记虚拟地址;可通过确保没有映射到页的虚拟地址允许拒绝访问来防止所选择的到页的访问(如,写)。地址翻译映射的这一方面可用于实现一种形式的存储器保护。由此,通过确保向软件对象展现的任一映射处于没有映射到所讨论的页的虚拟地址准许访问的状态中,使该软件对象(如,操作系统、应用程序级进程或任一类型的软件对象)可以拒绝到物理地址空间的页的地址。这一类型的存储器保护模式在 IA32 处理器家族(如,INTEL x86 处理器)中尤其有用,因为 INTEL x86 处理器体系结构是当在保护模式中(处理器的正常操作状态)操作时,所有存储器访问需要通过虚拟地址翻译。通过禁止超级用户模式程序以将准许对某些物理地址的某一访问的方式修改翻译表而其作用的存储器保护模式被称为“地址翻译控制”或 ATC。

[0004] 在典型的体系结构(如,x86)中,从虚拟地址到物理地址的翻译由普通存储器页(所谓的“页映射”页)的内容给出。这对于写操作系统是方便的,因为可由普通的存储器操作创建并修改虚拟地址映射。如果使用 ATC 配置操作系统,则 ATC 必须防止操作系统具有允许其直接写入页映射页的映射,因为操作系统将使用到这类页的写来创建映射,向任意物理存储器页给予任意的访问。由此,除防止对软件对象不允许写的页的读写映射之外,ATC 必须防止包括到页映射页的读写映射的“不安全”映射。

[0005] 尽管 ATC 的存储器隔离是有效的,引发的一个问题是如何处理创建不安全映射但是其本身不违反访问控制政策的写请求。处理这一写请求的一种方法是仅令该请求失败;然而,这将需要对操作系统实质上的修改。由此,一些当前的 ATC 算法要么修改写值(如,将到页映射页的读写映射改变为只读映射),要么修改其它页映射页来令该映射变得安全。该技术的问题是软件对象将执行该写请求,相信正在向目标位置写入指定的值,然而实际上,该位置最终将包含不同的值。这一偏差将以各种方式产生反响-如,软件对象可基于软件认为已储存的值来生成校验和,并且这些校验和对于由 ATC 系统生成的修改值是无效的。

[0006] 本发明的一个实施例的一个好处是提供一种环境,其中创建不安全映射(但是遵

循安全政策)的写看似成功地未修改(从软件对象的观点来看),但是所得的映射无法被利用来回避安全政策,由此克服了现有技术的缺点。

发明内容

[0007] 本发明提供了阴影页的使用来支持地址翻译控制。在典型的虚拟寻址系统中,给定的页可以是映射页(包含作为映射的一部分的数据)或数据页(某一虚拟地址的目标),或两者。映射页和数据页在不同的上下文环境中访问;数据页中的条目是基础读或写请求的目标;另一方面,映射页中的条目被解除参照来查找其它页。本发明维护了对应于在其中可使用页的不同上下文环境的多个页副本(如,作为目录、表或数据页,如后文所解释的)。取决于页被访问的上下文环境,本发明使用合适的页副本。

[0008] 某些虚拟寻址系统(如 INTEL x86 处理器家族中使用的最常见的虚拟地址模式)具有两种类型的映射页:目录和表。目录包含到表和大数据页的引用,表包含到小数据页的引用。(“大”和“小”页在下文更具体描述。)由此,从地址翻译过程的观点来看,有多达三种可访问给定页的不同上下文环境:作为目录、作为表或作为目标数据。在一个较佳的实施例中,维护了给定页的多达三个版本:目录版本、表版本和数据版本。当访问给定页时,使用该页的目录、表或数据,取决于访问该页的上下文环境。

[0009] 在 ATC 下,仅当页用作目录或表时,页内容可导致存储器访问政策的违反。例如,页可包含到界限外页的链接。然而,仅当地址翻译器实际上将页用作映射的一部分时,才存在该页将用于访问界限外页的危险;如果相反该页作为数据页来访问,则它不暴露界限外页的虚拟地址。由此,页的数据副本可包含软件对象相信它已写入页的实际数据,而页的目录和表副本可包含给出安全映射的修改版本。

[0010] 下文描述了本发明的其它特征。

附图说明

[0011] 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:

[0012] 图 1 是实现本发明的各方面的示例计算环境的框图;

[0013] 图 2 是示例虚拟地址系统的框图;

[0014] 图 3 是地址翻译映射中示例条目的框图;

[0015] 图 4 是用于通过地址翻译控制的存储器访问控制的示例不变式条件的框图;

[0016] 图 5 是在多个版本中存在的页的框图,其中,取决于使用该页的上下文环境采用页的不同版本;

[0017] 图 6 是用于衍生页的目录版本的示例过程的流程图;以及

[0018] 图 7 是用于衍生页的表版本的示例过程的流程图。

具体实施方式

[0019] 综述

[0020] 地址翻译控制可用于通过有效地拒绝将违反政策的虚拟地址映射来实现存储器

访问控制政策。一般而言,地址翻译控制通过在编辑地址翻译映射的尝试上施加实质上的限制来起作用,使得该映射总是保持安全(在该映射不向给定的软件实体暴露到该实体的界限外(或不可写)的页的链接(或可写链接),并且不允许不可信实体写入该映射的意义上)。通常,通过评估编辑映射的请求以确定实施该请求是否会将该映射置于不利状态,来施加这些实质上的限制;如果将导致不利状态,则修改该请求使得实施所期望的请求仍能维持所期望的状态。(如,可修改将导致向政策下可读但不可写的页的读/写链接的编辑映射的请求以将该链接标记为只读。)该技术的问题是有时候软件的正确行为取决于包含该软件认为它已写入存储器的值的存储器—如,在核实校验和的情况下—并且修改请求将导致存储器包含不同于软件认为它所写入的值的值。本发明通过维护用作页映射页的页的不同版本解决了这一问题:向软件对象展现的数据版本,以及可以在不违反映射安全的情况下用作地址翻译过程的一部分的一个或多个映射版本。令这一页的数据版本变得只读,使得可由 ATC 截取对页的写,从而可修改页使得保持不同的版本同步。

[0021] 示例性计算方案

[0022] 图 1 示出了适合在其中实现本发明的各方面的一个示例计算环境。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。

[0023] 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。适合使用本发明的已知的计算系统、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、嵌入式系统、包括任一上述系统或设备的分布式计算环境等等。

[0024] 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,如存储器存储设备。

[0025] 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 可代表多个逻辑处理单元,如多线程处理器上所支持的。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 也可被实现为点对点连接、交换光纤等通信设备。

[0026] 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、CD-ROM、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。

[0027] 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。

[0028] 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。仅作为示例, 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。

[0029] 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。

[0030] 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。

[0031] 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

LAN171。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 WAN173,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。

[0032] 示例性虚拟地址模式

[0033] 图 2 示出了虚拟地址系统的一个示例。图 2 描述的示例是一分页类型虚拟地址模式,尽管可以理解,虚拟寻址可基于其它模型,如分段(segmentation)。图 2 所示的模式是二级地址模式,如 INTEL x86 处理器上可用的虚拟寻址模式之一。该模式是“两级”的,因为必须使用两级“间接”方将虚拟页标识符翻译成物理页,如后文所述。

[0034] 在该分页模式中,页目录 202 包含一组条目。条目的一个示例结构在下文结合图 3 更详细地描述,但是本质上,每一条目标识了具体页表,如页表 204(1)、204(2) 或 204(3) 的物理位置(如,页帧号即“PFN”)。每一页表进而包含一组条目,每一条目标识具体数据页,如页 206(1)、206(2)、206(3) 或 206(4) 的物理位置(还是页帧号)。数据页是 RAM132 的确定长度的连续部分。数据页可储存任一类型的数据,并且应当注意,除储存普通数据之外,数据页也可用于储存页目录 202 和页 204(1) 到 204(3) 的内容。由此,给定的页可以是目录、表、数据页或作为这三个结构的组合扮演多重角色。

[0035] 图 2 描述的虚拟地址模式是两级虚拟地址模式,因为需要同时通过页目录(级 1)和页表(级 2)以查找具体页。本领域的技术人员可以理解,可以用任意级数来设计虚拟地址系统,并且本发明的原理可应用到所有这类虚拟地址模式。如本领域所已知的,INTEL x86 处理器支持具有一、二或三级的虚拟地址,并且通常采用一种“混合”模式,其中“小”页(即,长度为 4 千字节的页)使用两级虚拟地址,“大”页(即,长度为 4 兆字节的页)使用一级虚拟地址。

[0036] 在图 2 的分页模式中,页上的一个字节可由虚拟地址 210 来标识,包括页目录偏移 211、页表偏移 212 和页偏移 213。(该虚拟地址结构应用到储存在小页中的数据;大页在后文描述。)由此,为查找一物理地址,执行地址翻译的存储器管理单元(MMU)220 使用页目录偏移 211 来查找页目录 202 中具体条目。例如,偏移 211 可等于零,指示应当参考页目录 202 中的第零个条目。该条目包含储存页表的 PFN,因此 MMU20 使用该 PFN 来查找页表之一(如,页表 204(1))。MMU220 然后使用页表偏移 212 作为到标识的页表的索引,并检索在该偏移处找到的条目。该条目包含数据页(如页 206(1))的 PFN,因此 MMU220 将页偏移 213 添加到标识的页的基础地址,以查找物理存储器的具体字节。除纯地址翻译之外,MMU220 也可适用于执行各种其它功能:如,如果表中的页条目被标记为“不存在”,则 MMU220 可从盘加载该页;如果该页被标记为“只读”,则 MMU 可不允许写访问,等等。

[0037] 如果虚拟地址涉及大页,则虚拟地址的结构以及翻译该地址的过程与上述有些许不同。虚拟地址仅包含一个偏移,为到目录的索引。在该偏移处查找的目录条目不是包含页表的 PFN,而是包含大数据页的 PFN。该目录条目也具有一个位,它被设置成指示该条目涉及大页而非页表。当设置了大页位时,虚拟地址不包含到页表的索引,因此在翻译过程中不使用页表;相反,虚拟地址的剩余部分(如,除到目录的索引之外的部分)被作为到大页

的索引来处理。绕过了页表级,因此仅发生一级翻译。

[0038] 在图 2 的虚拟地址模式中,页目录本身的位置(即,PFN)储存在存储位置 201。当开始翻译虚拟地址 210 时,MMU220 使用存储位置的内容来查找页目录 202。由此,可以存在多个页映射,并且可通过设置存储位置 201 的内容包含给定映射的页目录的 PFN 来选择具体的映射用于当前使用。在 INTEL x86 处理器的示例中,存储位置 201 对应于名为 CR3 的寄存器。

[0039] 如上述者,页表或页目录中的每一条目包含具体物理页的 PFN,并也可包含某些其它数据。图 3 示出了页表或页目录中条目 300 的示例结构。

[0040] 条目 300 包括具体物理页的 PFN302。例如,如果条目 300 是页目录的一部分,则 PFN302 是应当在下一级地址翻译过程中被参考的页表的 PFN(或者,在大页链接的情况下,条目仅包含该条目所参考的大数据页的 PFN)。另外,条目 300 包含一个位 304,指示该条目是用于大页还是小页。(仅当条目 300 为页目录的一部分时该位才有意义。为简洁计,条目的格式可以相同,无论该条目是目录的一部分还是表的一部分,尽管在表的情况下,位 304 的意义未下定义。)

[0041] 只读位 306 指示作为条目的最终目标的数据页是否被视为读/写(位清除)或只读(位设置)。如果目标数据页是只读的,则对该页的写请求将失败。(MMU220 可用于强迫页的只读状态。)只读位在目录和表条目中皆可存在;如果最终导向数据页的目录和表链接各自的只读位设置发生冲突,则可使用冲突解决规则来确定该目标数据页是读/写还是只读。例如,冲突规则可规定,为将页作为读/写处理,通向该页的目录和表链接必须都被标记为读/写(即,两者条目中的只读位 306 必须被清除)。同一页可由通过映射的不同路径到达,并且是被作为读/写还是只读处理要取决于使用了哪一路径来达到该页。

[0042] 存在位 308 指示目标数据页当前是在物理存储器中存在,还是需要从盘上复制到存储器。例如,如果存在位 308 被清除(指示目标页不存在),则对该页的访问请求将生成页错误,该错误然后由中断服务例程处理,将页的内容从盘复制到物理存储器并调整地址翻译映射来反应页的物理位置。当对于给定映射,如果页目录和页表条目中的存在位被不同地设置时,这些位之间的冲突可由类似于上文对读/写位所讨论的冲突解决规则来分解—如,当且仅当目录和表条目都被标记为存在时,该映射被视为存在。

[0043] 使用地址翻译表条目编辑控制(ATC)的存储器访问控制

[0044] 结合图 2-3 所描述的虚拟地址模式的一个特征是有可能存在部分物理存储器,它没有对应的虚拟地址。这一观察的必然结论是,给定存储器的任一部分,可通过确保该地址翻译映射不通向该存储器部分来限制对该存储器部分的访问。实际上,由于存储器位置没有虚拟地址,该存储器位置被呈现为界限外。(在许多系统中—如,INTEL x86 处理器—几乎所有的存储器访问请求由虚拟地址作出;对于由物理地址作出访问请求的那些限制的情况,可使用并行访问控制机制。)

[0045] ATC 用于实现存储器访问控制机制的方法按照以下符号来解释:NA(“无访问”)是在某一政策下不允许访问的页的组。MP(“映射页”)是可通过地址翻译映射访问的页的组—即,存在虚拟地址的页的组。只要条件 $NA \cap MP = \phi$ 连续成立,则防止了对作为 NA 成员的页的访问。该条件在图 4 中描述为维恩(Venn)图:即,页 406 是机器上可用的物理页组;MP402 是存在虚拟地址的页组;NA404 是在政策下不允许访问的页组。只要图 4 的条件

连续成立,即不可使用虚拟地址来访问在政策下处于界限外的页,因为该映射不通向这些页(即,没有虚拟地址的页)。由此,图4描述的条件可用于通过ATC实现存储器访问控制。该条件被称为“不变式”,因为ATC的目标是限制对地址翻译映射的改变以防止条件从为真的状态改变为非真状态。

[0046] 图4描述了可用于存储器访问控制的简单不变式条件,并仅为说明目的示出。取决于访问控制发生的环境,可有更复杂的条件。例如,可设置关于哪些条目被包含在目录(或表)中的具体规则,并通过使用只读和/或存在位来帮助强迫访问控制条件来改善访问控制。例如,以下描述是可用于在INTEL x86处理器上通过ATC实现存储器访问控制的一组规则:

[0047] D1是可用作页目录的页组。D2是可用作页表的页组。 $D = D1 \cup D2$ 。页目录或页表中被标记为“存在”的每一条目(即,其存在位被设置)被称为“链接”。如果有从D1中的某一页到D2页的小读写链接,则所讨论的D2页是“写活动的”。“小”链接是从目录到表的链接—即,目录中最终将通向小页的链接。“大”链接是目录中指向大页的链接。)假定有这样的政策来定义允许某一实体对其进行读和/或写访问页。

[0048] 维护以下不变式:

[0049] -CR3在D1中;

[0050] -所有D1和D2页在相关政策下是可读的;

[0051] -D1页的每一小链接指向一个D2页;

[0052] -D2页的链接指向在政策下可读的页;

[0053] -写活动的D2页的每一读写链接指向在政策下可写且不在D中的页;

[0054] -包含在从D1页的大链接的大页目标中的每一小页在政策下是可读的;如果该链接是读写的,则小页在政策下也可写,且不在D中。

[0055] ATC被用来确保防止那些将破坏上述不变式的对地址翻译映射的变化。维护这些不变式确保了所讨论的实体不能破坏政策。

[0056] 不管施加了哪一不变式,不变式的真实性可通过评估每一访问请求来确定假如实际实现了该请求,它是否会导致不变式继续成立的状态来维护。如果所得的状态将满足该不变式,则实现该请求。然而,如果该不变式将不再成立,则至少有两个选项:

[0057] -(1) 拒绝该请求;或

[0058] -(2) 将请求修改成将继续满足该不变式的形式。

[0059] 选项(1)有缺点,在实践中,将需要拒绝大量访问请求,这将破坏计算机系统的功能。然而,采用选项(2),软件对象将值写入存储位置,并且该存储位置最终储存不同于软件对象认为它所写的值的修改值;如上所述,软件的正确功能(如,校验和核实)将取决于储存软件认为它所写入存储器的实际值的存储器,并且由此,选项(2)也可破坏软件的功能。本发明通过储存页的多个版本解决了这一问题。一个版本包含程序认为它写入该页的确切数据。页的另一副本包含其如果在地址翻译过程中使用将促使维护相关不变式条件的数据的版本。

[0060] 阴影页

[0061] 依照本发明的一个特征,页可有多表示法。同一页的多个表示法确保了有包含程序实际上写入页的数据的页的版本,以及在地址翻译过程中可安全地用作页目录和页表

的页的其它（“阴影”）版本。本上下文中的“安全使用”是指将阴影页用作目录（或表，视具体情况而定）将不导致违反应用到 ATC 系统的不变式。

[0062] 较佳地，给定页 x ，有该页的三个版本，应当被称为 $d(x)$ 、 $t(x)$ 和 $m(x)$ 。 $d(x)$ 是页的“目录”版本 – 即，适合用作上述地址翻译过程中的页目录的页的版本。 $t(x)$ 是安全地用作页表的页的版本。 $m(x)$ 是页的“存储器”版本 – 即，包含由一个或多个程序写入页的实际数据的版本。在本发明的描述中，术语“ $d(x)$ ”要么是指页 x 的目录版本的内容，要么是指储存页 x 的目录版本的 PFN。对于 $t(x)$ 和 $m(x)$ 也是一样。术语 $d(x)$ 、 $t(x)$ 和 $m(x)$ 是否涉及页内容或其 PFN 可从上下文清楚，或可特别指明。

[0063] 图 5 示出了 $d(x)$ 、 $t(x)$ 和 $m(x)$ 如何在地址翻译过程中使用。图 5 涉及三个页，标签为 x 、 y 和 z （分别为标号 502、502 和 506）。页 x 是页目录，页 y 是页表，页 z 是数据页。应当注意，页 x 、 y 和 z 可扮演多重角色。由此，页 x 可担当页目录或数据页，视具体情况而定。页 y 在某一上下文中可担当页表，而在其它上下文中担当页目录。然而，为图 5 的目的而假定正在翻译具体的虚拟地址，并且为翻译该地址的目的，页 x 、 y 和 z 分别扮演目录、表和数据页的角色。

[0064] 页 x 以版本 $d(x)$ 、 $t(x)$ 和 $m(x)$ 存在。为翻译所讨论的地址，参考页 x 的目录版本（即， $d(x)$ ）。作为目录， x 中的条目包含目标页表的 PFN，并且 x 和 $d(x)$ 之间的原则差异是对于页目录 x 中的每一目标 t_i ， x 包含 t_i 的 PFN，而 $d(x)$ 包含 $t(t_i)$ 的 PFN – 换言之， $d(x)$ 被修改成指向目标页的表版本而非那些页的原始版本。

[0065] 当参考页 $d(x)$ 时， $d(x)$ 中的相关条目（即，由虚拟地址的目录偏移部分指示的条目，图 2 中的元素 211）指向页 $t(y)$ 。然后参考页 $t(y)$ 来查找具体数据页。 $t(y)$ 中的条目包含目标数据页的 PFN。 $t(y)$ 和 y 之间的关系类似于 $d(x)$ 和 x 之间的关系：对于由 y 引用的每一数据页 d_i ， $t(y)$ 包含 $m(d_i)$ 的 PFN 而非 d_i 的 PFN。（然而，应当注意， $m(d_i)$ 的 PFN 一般与 d_i 的 PFN 相同，因为在页的原始位置储存页的数据版本一般是存储器的最有效使用。）使用由虚拟地址的表偏移域所指示的偏移（如，图 2 中的元素 212），查找表 $t(y)$ 中合适的条目。该条目涉及具体数据页的 PFN，本示例中为 $m(z)$ 。

[0066] 在页 $m(z)$ 被标识之后，基于虚拟地址中所指示的页偏移（如，图 2 中的元素 213）访问页 $m(z)$ 中合适的数据单元。

[0067] 由此，在常规的地址翻译过程中，到数据页的路径从页 x 通向页 y 再到页 z 。当使用了依照本发明的阴影页时，翻译页从页 $d(x)$ 通向页 $t(y)$ 再到页 $m(z)$ 。

[0068] 创建页 $d(x)$ 、 $t(x)$ 和 $m(x)$

[0069] 通过在页 x 上执行定义的变换来创建页 $d(x)$ 、 $t(x)$ 和 $m(x)$ 。以下是这些变换的较佳实施例的描述。

[0070] 较佳地， $m(x)$ 表示程序写入页 x 的实际数据，没有任何类型的修改或过滤。换言之，从 x 到 $m(x)$ 的变换本质上是相同的变换。

[0071] 较佳地，依照以下规则创建 $d(x)$ 和 $t(x)$ 。 $d(x)$ 与 x 相同，但是对于涉及被标记为存在的页 x 的条目的每一页 t_i ， $d(x)$ 中对应的条目涉及 $t(t_i)$ 的 PFN 而非 t_i 的 PFN。 $t(x)$ 与 x 相同，但是对于涉及被标记为存在的页 x 的条目的每一页 d_i ， $t(x)$ 涉及 $m(d_i)$ 的 PFN 而非 d_i 的 PFN。因此，如果在相关政策下，目标页是可读但不可写，或如果目标页是页目录或页表，则该条目被标记为只读。

[0072] 以下是如何创建 $d(x)$ 和 $t(x)$ 的更正式描述。为描述之目的, $D1$ 是可用作页目录的 PFN 的组, $D2$ 是可用作页表的 PFN 的组。命题 (statement) $D1.x$ 是指 x 是 $D1$ 的成员, $D2.x$ 指 x 是 $D2$ 的成员。设 M 是存储器映射, 如可通过在其上实施存储器访问控制模式的相关软件对象所见到的。 $M.x.e$ 表示储存在其 PFN 为 x 的物理页的第 e 个条目中的值。 $R.x$ 指在 x 在相关政策下是可读的, $W.x$ 指 x 在相关政策下是可写的。 m, t, d 和 P 令以下成立 (在每一情况下, 设 b 是 $M.x.e$, 并设 $D.x = D1.x \vee D2.x$) :

[0073] - 如果 $\neg R.x$, $m.x = t.x = d.x =$ 未定义, 其中, “未定义”是不在物理存储器中的页的 PFN。

[0074] - $\neg d.x =$ 如果 $D1.x$, 则为 x , 否则未定义

[0075] - $\neg D2.x \Rightarrow t.x =$ 未定义

[0076] - $P.(m.x).e = v$ (即, 在 P 中, $m.x$ 看似与 x 在 M 中的完全相同)

[0077] - $d.x \neq \text{未定义} \Rightarrow P.(d.x).e =$ 如果 v 存在, 则为 $v[pfn \leftarrow t.(v.pfn)]$, 否则为 v

[0078] - $t.x \neq \text{未定义} \Rightarrow P.(t.x).e =$ 如果 v 存在, 则为 $v[pfn \leftarrow m.(v.pfn),$
 $rw \leftarrow (v.rw \wedge (R.(v.pfn) \Rightarrow W.(v.pfn)) \wedge \neg D.(v.pfn))]$, 否则为 v

[0079] 换言之, x 的目录版本看似与 x 的存储器版本完全相同, 但是其 PFN 重定向到表版本; x 的表版本看似与 x 的存储器版本完全相同, 但是其 PFN 重定向到读版本, 并且对可读但不可写或在 D 中的目标清除读写位 (或对此类目标设置只读位, 取决于条目的表示是包括读 / 写位还是只读位)。注意, 对于目录, 在 x 处保留 x 的目录版本, 但是对于表, 在 x 处保留 x 的读版本以将从 $D2$ 中移除 x (如, 将其交换到盘) 的成本最小化。只要版本恰巧保留了同一数据 (如, 对于 ATC 允许的目录和表), 它们就共享同一物理页, 因此对于不试图作出 ATC 将拒绝或修改的写的软件对象, 不需要创建阴影页。

[0080] 图 6 和 7 分别示出了创建 $d(x)$ 和 $t(x)$ 的示例过程。

[0081] 现在参考图 6, 假定有作为映射 (如, 上述的映射 M) 的一部分的页 x , 并且期望基于 x 创建页 $d(x)$ 。首先, 确定 (602) 在相关政策下页 x 是否可读。如果 x 不可读, 则 $d(x)$ 未定义 (606), 并且过程终止。如果 x 可读, 则确定 (604) x 是否是 $D1$ 的成员 - 即, x 是否被指定为用作页目录。如果 x 不是 $D1$ 的成员, 则 $d(x)$ 未定义 (606)。如果 x 是 $D1$ 的成员, 则创建页 $d(x)$ 使它包含与页 x 相同的内容, 除条目中被标记为存在的的 PFN 域被改为指向其目标的表版本之外。可通过设置 $n = 0$ (607), 然后对于 n 的给定值, 确定 x 中的第 n 个条目是否被标记为存在 (608), 来达到该结果。如果 x 中第 n 个条目不被标记为存在, 则 $d(x)$ 中的第 n 个条目可被设置成等于 x 中第 n 个条目 (610)。如果 x 中的第 n 个条目被标记为存在, 则 $d(x)$ 中的第 n 个条目可被设置成等于 x 中第 n 个条目, 除其 PFN 被改为指向表版本 (612) 之外 - 即, 如果 $P.n.pfn$ 涉及页 P 的 PFN 域, 并且如果 v 具有上述意义, 则 $d(x).n.pfn = t(v.pfn)$ 。在设置了 $d(x)$ 中的第 n 个条目制售, 递增 n (614), 过程返回到 608 来设置下一条目。

[0082] 现在参考图 7, 再次假定有作为映射的一部分的页 x , 并且期望创建页 $t(x)$ 。最初, 确定 (702) 在相关政策下 x 是否可读。如果 x 不可读, 则将 $t(x)$ 设为未定义值 (706), 过程终止。如果在相关政策下 x 可读, 则确定 (704) x 是否是 $D2$ 的成员。如果 x 不是 $D2$ 的成

员,则将 $t(x)$ 设为未定义 (706)。如果 x 是 D2 的成员,则创建页 $t(x)$ 使得条目中的值等于 x 中的值,除被标记为存在的页的 PFN 被调整为指向目标页的存储器版本,并且调整某些读/写链接令其变为只读之外。为创建页 $t(x)$ 的内容,最初将计数器设为 0 (708)。然后确定页 x 中的第 n 个条目是否被标记为存在。如果该条目不被标记为存在,则将 $t(x)$ 中的第 n 个条目设为等于 x 中的第 n 个条目 (712)。如果第 n 页被标记为存在,则将页 $t(x)$ 中的第 n 个条目设为等于 x 中的第 n 个条目,除该条目的 PFN 域被设为指向目标页的存储器版本 (714) 之外—即,如果 x 中第 n 个条目的目标页具有 $PFN = A$,则 $t(x)$ 中第 n 个条目的 PFN 域被设为等于 $m(A)$ 。(如上所述, $m(A)$ 的 PFN 经常等于 A 的 PFN。)下一步,确定 (716) 第 n 个条目的目标页在相关政策下是否是可读但不可写的页。如果目标页可读但不可写,则将 $t(x)$ 中第 n 个条目标记为只读 (720)。否则,确定 (718) x 中第 n 个条目的目标页是否为 D1 或 D2 的成员。如果是,则 $t(x)$ 中第 n 个条目被标记为只读 (720)。然后递增计数器 n ,过程循环回至 714 以创建 $t(x)$ 中的下一条目。

[0083] 储存阴影页

[0084] 能够储存每一页的三个版本 (即, $d(x)$ 、 $t(x)$ 和 $m(x)$), 尽管有某些优化允许更有效地来表示页。首先,如果上述算法将导致创建与已储存的版本相同的阴影页,则不需要创建阴影页。由此,对于大多数页,仅需储存该页的一个版本,在这一情况下,对于这样的页 x , $d(x)$ 、 $t(x)$ 和 $m(x)$ 的 PFN 都相同。

[0085] 第二,对于所有不同于目录的页,页的原始版本较佳地担当该页的数据版本。由此,对于非目录页 (即,其帧号不是 D1 的成员的页), $m(x)$ 的 PFN 等于 x 的 PFN; x 的目录和表版本将储存在别处。然而,在目录页的情况下,该页的原始版本较佳地 (并且在一些情况下,必须) 担当该页的目录版本。由于按照允许担当目录的页 (如,在 INTEL x86 处理器上,其 PFN 可被加载到 CR3 的页) 的 PFN 来定义组 D1,将目录版本移到另一 PFN 是不实用的;由于 CR3 需要由可能不知道阴影页的存在的软件对象加载 (如,CR3 可由其存储器访问被 ATC 系统所限制的操作系统加载),该目录页可能需要位于软件对象认为那些页所在之处的 PFN 处。

[0086] 将阴影页用于大页

[0087] 如上所述,INTEL x86 处理器 (以及各种其它处理器) 支持大页的应用,在这一情况下,将虚拟地址翻译成物理地址不涉及页表。为使大页在上述的阴影机制中起作用,可为大页创建阴影页表,即使该阴影页表不对应于任一真实的页表。由此,当页目录 x 包含大链接时,可创建该页的目录版本 (即, $d(x)$),它在条目中包含对应于 x 的大链接的小链接。该小链接指向阴影页表 (即 $t(x)$),它包含到组成大页的各个小页的链接。由于上文描述的阴影算法的部分涉及令某些页变为只读,将大页分裂成小页避免了必须将整个大页标记为只读页。如需要,作为大页的部分的个别小页可被标记为只读。(将大页标记为只读的缺点是每一对该页的写请求将生成异常,并且必须由更有特权的组件 (如,执行 ATC 的组件) 来评估,以确定写是否能够在不违背可应用的不变式的情况下发生。需要以这一方式处理对大页的每一写请求将显著地降低系统性能。)

[0088] 注意,上述示例仅为解释目的提供,并且不应当被解释为对本发明的限制。尽管参考各种实施例描述了本发明,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,尽管此处参考具体的装置、材料和实施例描述了本发明,本发明不意味着限

制在所揭示的细节上 ;相反,本发明延及如处于所附权利要求书范围之内内的所有功能上等效的结构、方法和用途。从本说明书的教导获益的本领域的技术人员在不脱离本发明的各方面的范围和精神的情况下,可对其作出各种修改和变化。

计算环境 100

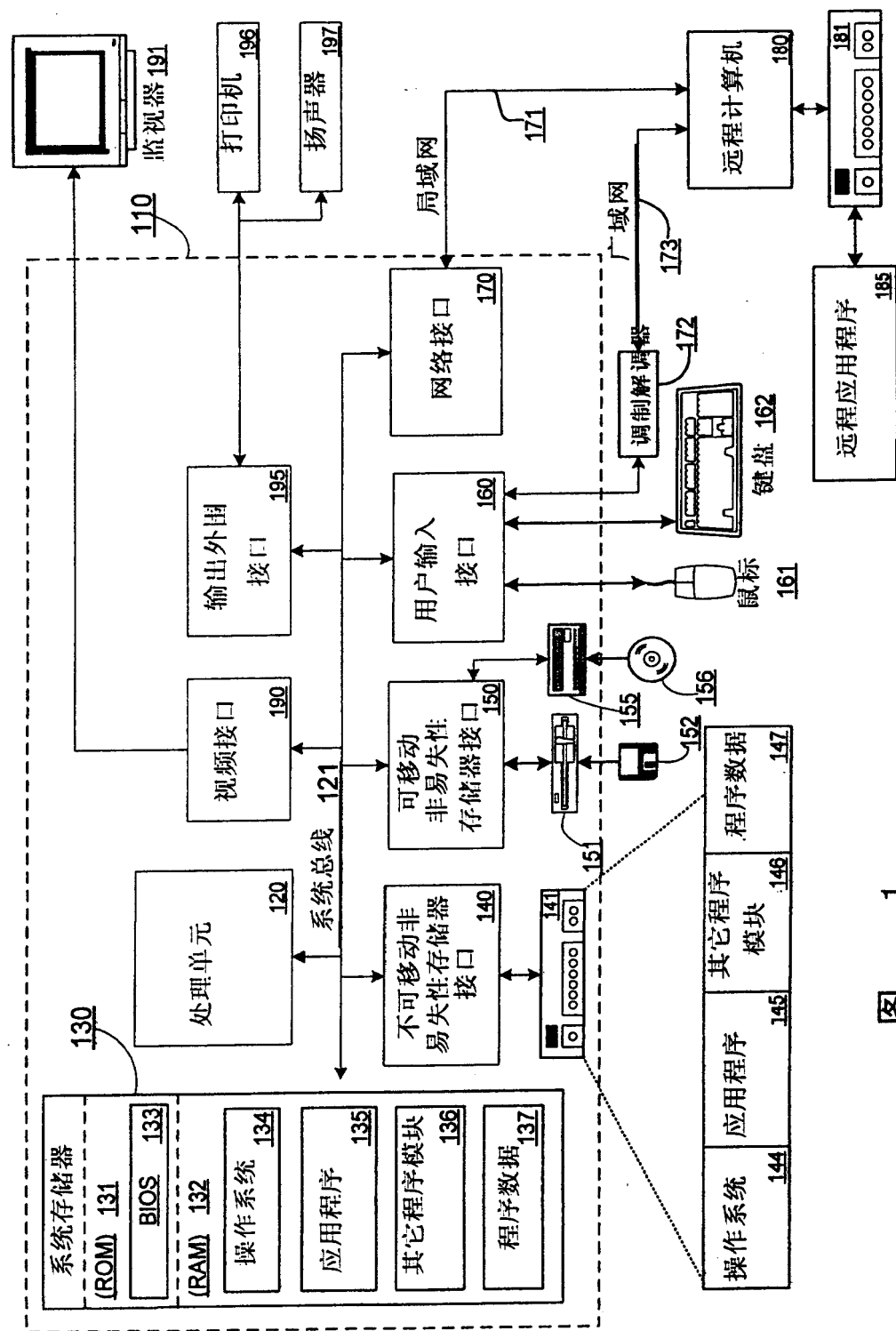
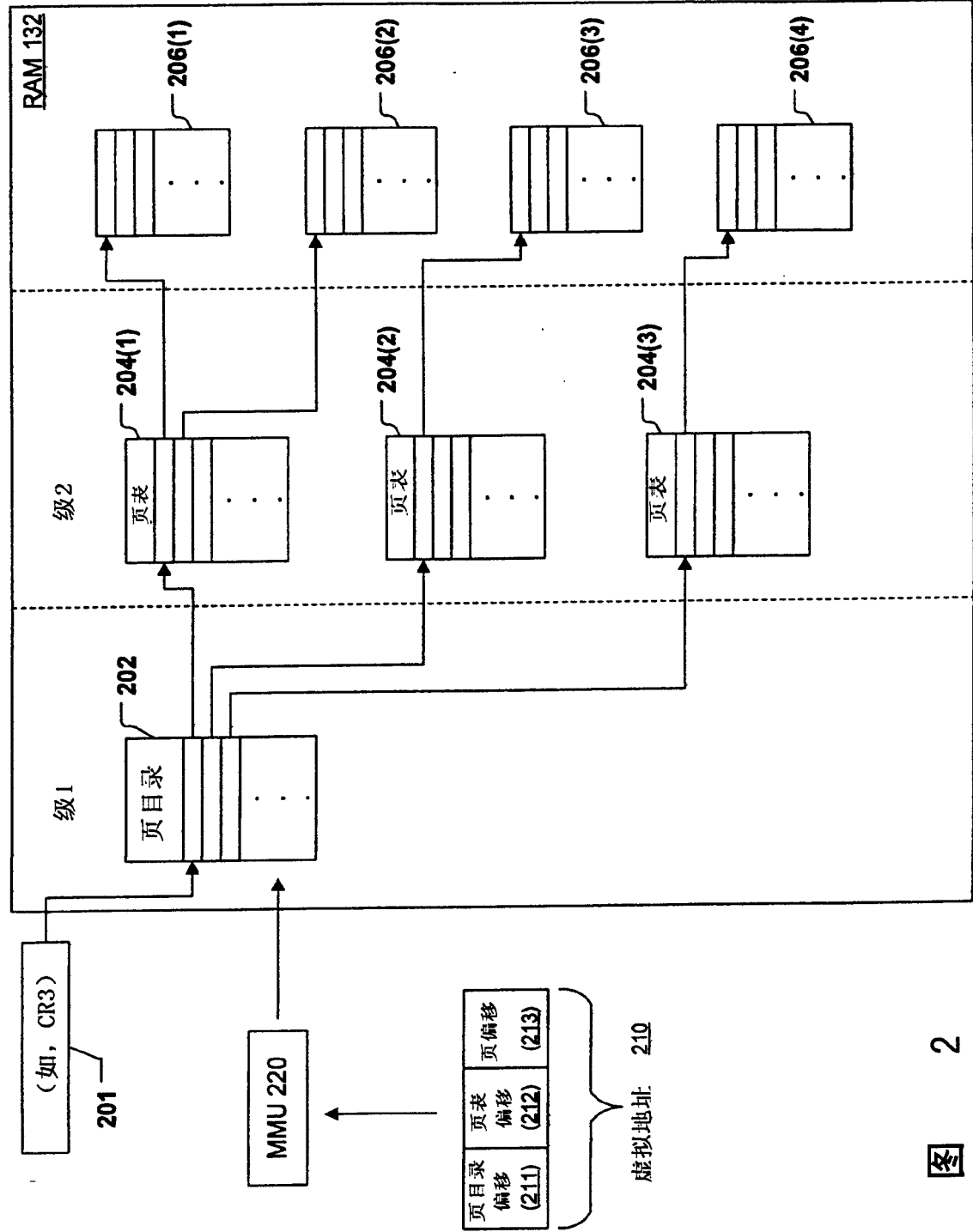


图 1



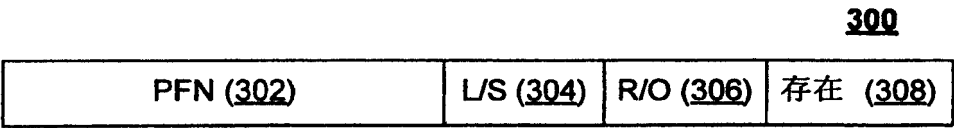


图 3

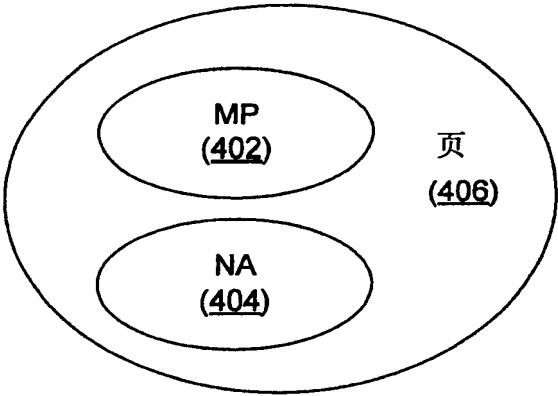


图 4

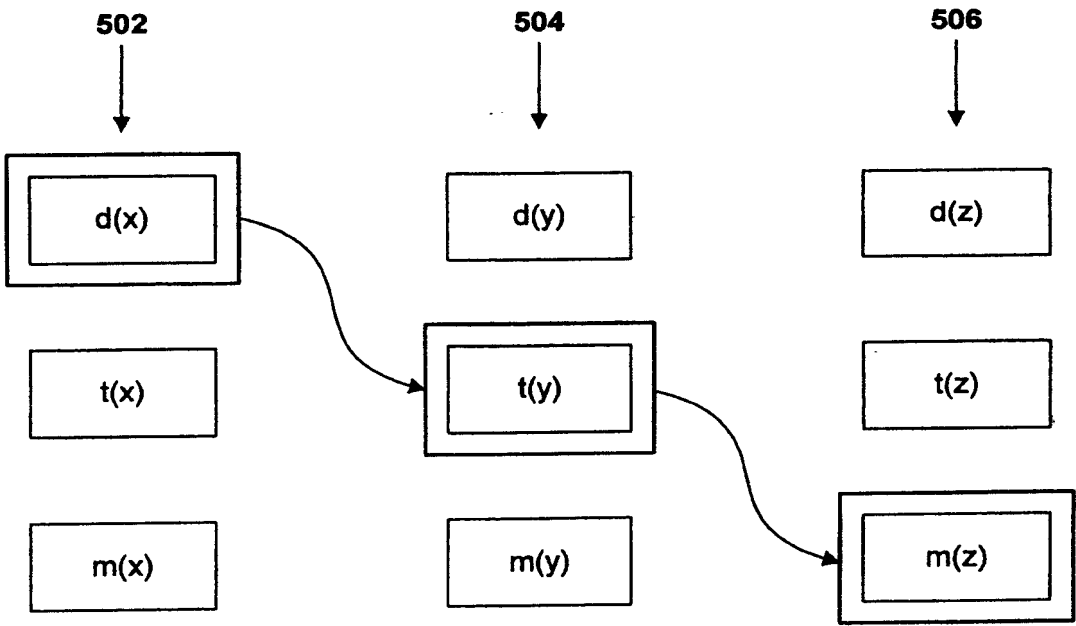


图 5

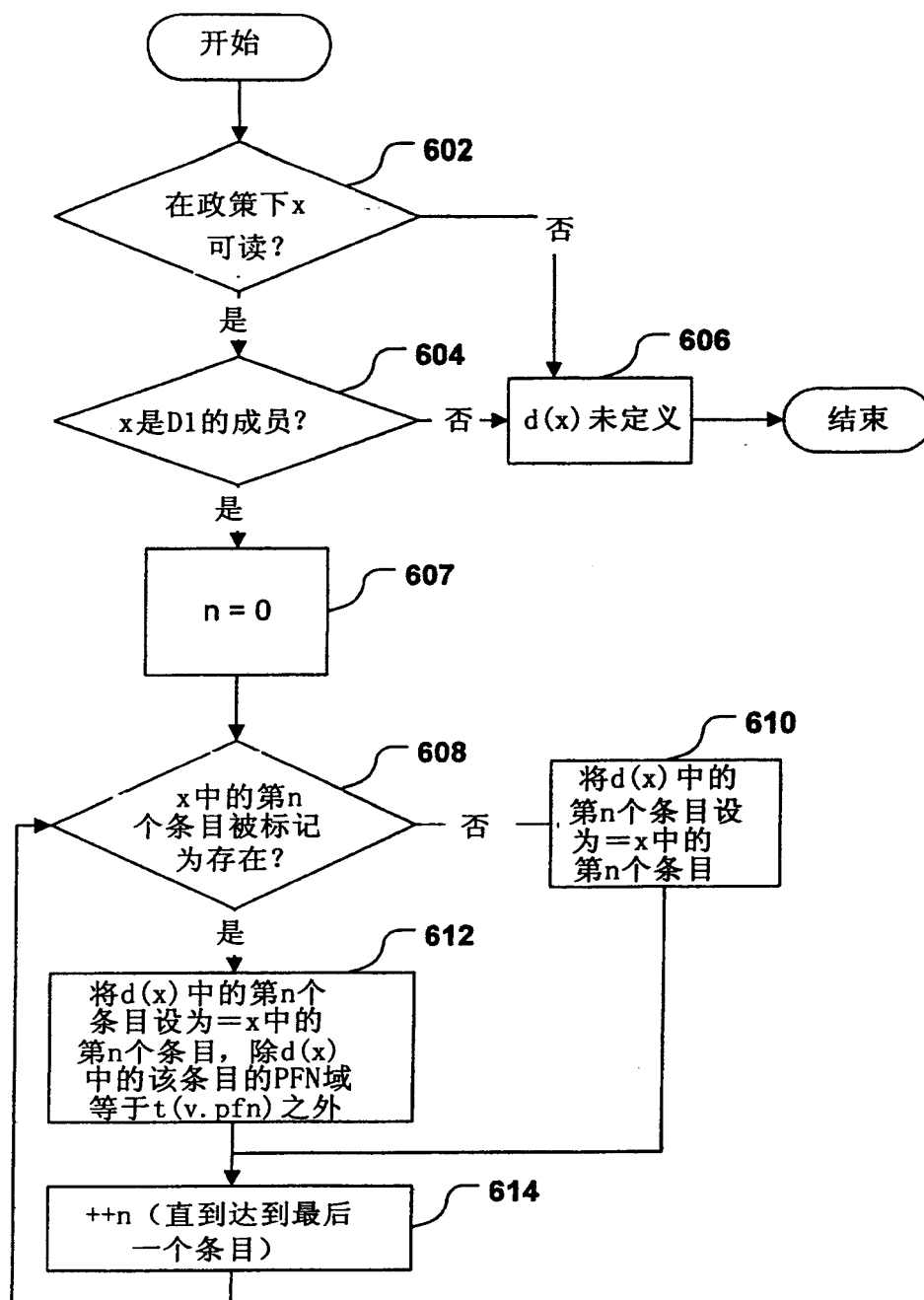


图 6

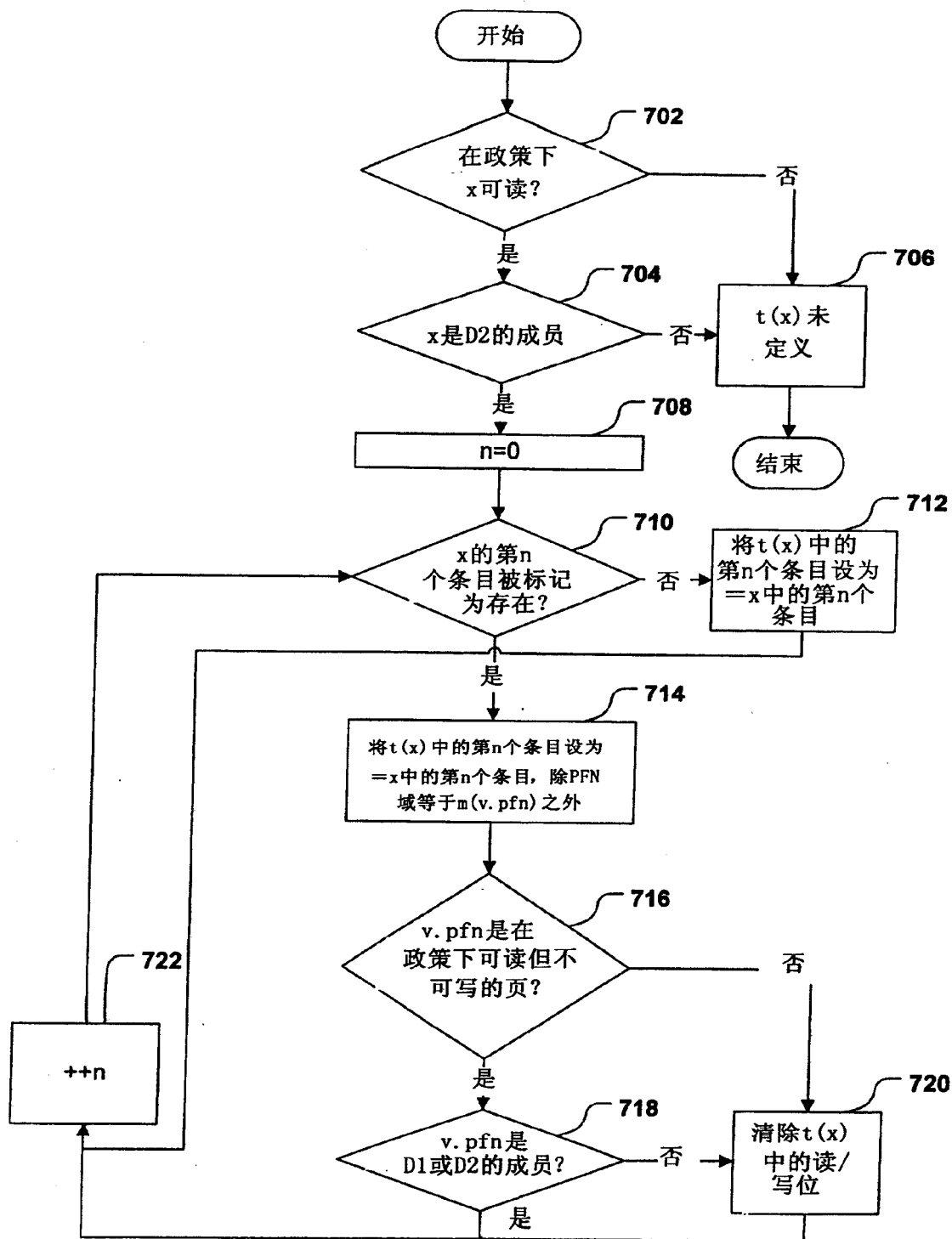


图7