República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(11) BR 112016004493-2 B1

(22) Data do Depósito: 12/09/2014

(45) Data de Concessão: 13/12/2022

(54) Título: MÉTODO, DISPOSITIVO DE COMPUTAÇÃO E MEIO DE ARMAZENAMENTO LEGÍVEL POR COMPUTADOR PARA IMPOSIÇÃO DE INTEGRIDADE DE CÓDIGO SELETIVA FACILITADA POR GERENCIADOR DE MÁQUINA VIRTUAL

(51) Int.Cl.: G06F 9/455; G06F 21/51.

(30) Prioridade Unionista: 12/02/2014 US 14/179,378; 17/09/2013 US 61/879,068.

(73) Titular(es): MICROSOFT TECHNOLOGY LICENSING, LLC.

(72) Inventor(es): DAVID A. HEPKIN; KENNETH D. JOHNSON.

(86) Pedido PCT: PCT US2014055290 de 12/09/2014

(87) Publicação PCT: WO 2015/041930 de 26/03/2015

(85) Data do Início da Fase Nacional: 29/02/2016

(57) Resumo: MÉTODO, DISPOSITIVO DE COMPUTAÇÃO E MEIO DE ARMAZENAMENTO LEGÍVEL POR COMPUTADOR PARA IMPOSIÇÃO DE INTEGRIDADE DE CÓDIGO SELETIVA FACILITADA POR GERENCIADOR DE MÁQUINA VIRTUAL. A presente invenção refere-se a um gerenciador de máquina virtual que facilita a imposição de integridade de código seletiva. Um gerenciador de máquina virtual (ou outra entidade privilegiada mais alta) pode verificar a integridade de código em páginas de memória, e um processador virtual que executa no modo de núcleo executa o código em uma página de memória somente se o gerenciador de máquina virtual (ou outra entidade privilegiada mais alta) verificou a integridade de código daquele código. No entanto, o gerenciador de máquina virtual não precisa verificar a integridade de código em páginas de memória quando o processador virtual está executando em modo de usuário. Ao invés, um sistema de operação que executa no processador virtual pode aplicar qualquer uma de uma variedade de políticas (por exemplo, opcionalmente executar qualquer uma de uma variedade de diferentes controles ou verificações do código) para determinar se o código pode ser executado no modo de usuário.

Relatório Descritivo da Patente de Invenção para **“MÉTODO, DISPOSITIVO DE COMPUTAÇÃO E MEIO DE ARMAZENAMENTO LEGÍVEL POR COMPUTADOR PARA IMPOSIÇÃO DE INTEGRIDADE DE CÓDIGO SELETIVA FACILITADA POR GERENCIADOR DE MÁQUINA VIRTUAL".**

ANTECEDENTES

[001] Conforme a tecnologia de computação avançou, os dispositivos de computação tornaram-se crescentemente interconectados. Apesar desta interconexão prover muitos benefícios, esta não sem os seus problemas. Um tal problema é que os dispositivos de computação estão crescentemente expostos a programas maliciosos. Os programas maliciosos podem operar em diferentes modos, tal como roubando informações de um dispositivo de computação, desabilitando um dispositivo de computação, utilizando um dispositivo de computação para lançar ataque contra outros dispositivos de computação, e assim por diante. Apesar de algumas técnicas terem sido desenvolvidas para proteger um dispositivo de computação contra os programas maliciosos, tais programas maliciosos permanecem e podem levar a uma experiência de usuário frustrante quando estes infectam um computador.

SUMÁRIO

[002] Este Sumário está provido para introduzir uma seleção de conceitos em uma forma simplificada que estão adicionalmente abaixo descritos na Descrição Detalhada. Este Sumário não pretende identificar características chave ou características essenciais do assunto reivindicado, nem pretende ser utilizado para limitar o escopo do assunto reivindicado.

[003] De acordo com um ou mais aspectos, uma página de memória que inclui um código executável a ser executado por um processador virtual de uma máquina virtual é identificada, a máquina virtual

sendo gerenciada por um gerenciador de máquina virtual. Uma determinação é feita quanto a se a página de memória deve ser executável em um modo de núcleo. Em resposta à determinação que a página de memória deve ser executável no modo de núcleo, uma verificação de integridade de código do código executável é executada e a execução do código executável é permitida para o modo de núcleo somente se a verificação de integridade de código verificar o código executável. Em resposta à determinação que a página de memória não deve ser executável no modo de núcleo, um sistema operacional da máquina virtual é permitido determinar se permitir a execução do código executável.

[004] De acordo com um ou mais aspectos, um dispositivo de computação inclui um sistema operacional, um gerenciador de máquina virtual, e um processador. O processador está configurado para permitir que o gerenciador de máquina virtual restrinja a execução de modo de núcleo de páginas de memória para as páginas de memória tendo um código a integridade do qual foi verificada por uma entidade privilegiada mais alta que é mais privilegiada do que o sistema operacional, mas permite a execução de modo de usuário de páginas de memória sem observar se a integridade de código nas páginas de memória foi verificada pela entidade privilegiada mais alta.

BREVE DESCRIÇÃO DOS DESENHOS

[005] Os mesmos números são utilizados através de todos os desenhos para referenciar características iguais.

[006] A Figura 1 é um diagrama de blocos que ilustra um dispositivo de computação exemplar que implementa as técnicas aqui discutidas de acordo com uma ou mais modalidades.

[007] A Figura 2 ilustra um sistema exemplar que implementa as técnicas de imposição de integridade de código seletiva facilitadas por gerenciador de máquina virtual aqui discutidas de acordo com uma ou mais modalidades.

[008] A Figura 3 é um fluxograma que ilustra um processo exemplar para implementar a imposição de integridade de código seletiva facilitada por gerenciador de máquina virtual de acordo com uma ou mais modalidades.

[009] A Figura 4 ilustra outro sistema exemplar que implementa a imposição de integridade de código facilitada por gerenciador de máquina virtual de acordo com uma ou mais modalidades.

[0010] A Figura 5 ilustra um sistema exemplar que inclui um dispositivo de computação exemplar que é representativo de um ou mais sistemas e/ou dispositivos que podem implementar as várias técnicas aqui descritas.

DESCRIÇÃO DETALHADA

[0011] A imposição de integridade de código seletiva facilitada por gerenciador de máquina virtual está aqui discutida. Uma máquina virtual é uma implementação de software de um dispositivo físico que pode executar programas análogos a um dispositivo físico. A máquina virtual, e o acesso ao hardware do dispositivo físico, são gerenciados por um gerenciador de máquina virtual no dispositivo físico. A máquina virtual e o gerenciador de máquina virtual acessam uma memória que é composta de múltiplos blocos ou porções referidos como páginas de memória (ou simplesmente páginas). A integridade de código é utilizada para facilitar a proteção contra um código malicioso no dispositivo físico. A integridade de código refere-se a integridade de código (por exemplo, um binário) sendo verificada com base em uma política de integridade de código. Se o código for verificado com base na política de integridade de código, então a integridade do código é verificada e o código é permitido executar; de outro modo, a integridade do código não é verificada e o código não é permitido executar.

[0012] Um processador pode executar o código em um modo de núcleo ou em um modo de usuário. Quando um processador virtual da

máquina virtual está executando no modo de núcleo, o processador virtual executa somente um código a integridade do qual é verificada pelo gerenciador de máquina virtual (ou por outra entidade mais privilegiada que o sistema operacional que executa no processador virtual). A integridade de código de páginas de memória que incluem o código é verificada, e o processador virtual que executa no modo de núcleo pode executar o código em uma página de memória somente se o gerenciador de máquina virtual (ou outra entidade mais privilegiada) verificou a integridade de código do código na página de memória. No entanto, quando o processador virtual está executando no modo de usuário, um sistema operacional que executa no processador virtual determina se o código pode ser executado. O sistema operacional pode aplicar qualquer uma de uma variedade de políticas (por exemplo, executar qualquer uma de uma variedade de diferentes controles ou verificações do código) para determinar se o código pode ser executado no modo de usuário, incluindo opcionalmente executando o código no modo de usuário sem executar quaisquer controles ou verificações do código.

[0013] A Figura 1 é um diagrama de blocos que ilustra um dispositivo de computação exemplar 100 que implementa as técnicas aqui discutidas de acordo com uma ou mais modalidades. O dispositivo de computação 100 pode ser qualquer um de uma variedade de diferentes tipos de dispositivos. Por exemplo, o dispositivo de computação 100 pode ser um computador desktop, um computador servidor, um computador laptop ou netbook, um computador tablet ou notepad, uma estação de móvel, um aparelho de entretenimento, um decodificador comunicativamente acoplado um dispositivo de display, uma televisão ou outro dispositivo de display, um celular ou outro telefone sem fio, um console de jogos, um computador automotivo, um computador usável, e assim por diante.

[0014] O dispositivo de computação 100 inclui um gerenciador de máquina virtual 102, também referido como um hipervisor, e um ou mais componentes 104. O gerenciador de máquina virtual 102 gerencia o acesso pra a funcionalidade provida pelos componentes 104. Alternativamente, o gerenciador de máquina virtual 102 pode executar em um sistema operacional hospedeiro (não mostrado), em cujo caso o sistema operacional hospedeiro gerencia o acesso à funcionalidade provida pelos componentes 104.

[0015] Os componentes 104 podem ser uma variedade de diferentes componentes de processador, componentes de entrada / saída (I/O), e/ou outros componentes ou dispositivos. Por exemplo, os componentes 104 podem incluir um ou mais processadores ou núcleos de processador, um ou mais componentes de memória (por exemplo, memória volátil e/ou não volátil), um ou mais dispositivos de armazenamento (por exemplo, discos óticos e/ou magnéticos, unidades de memória instantânea), um ou mais componentes de comunicação (por exemplo, adaptadores de rede com fio e/ou sem fio), suas combinações, e assim por diante. Apesar de ilustrados como parte do dispositivo de computação 100, um ou mais dos componentes 104 (por exemplo, um ou mais dispositivos de armazenamento) podem ser implementados externos ao dispositivo de computação 100. Vários componentes ou módulos que executam no dispositivo de computação 100, incluindo o gerenciador de máquina virtual 102, podem acessar esta funcionalidade provida pelos componentes 104 diretamente e/ou indiretamente através de outros componentes ou módulos.

[0016] O gerenciador de máquina virtual 102 permite que uma máquina virtual 106 execute no dispositivo de computação 100. Uma única máquina virtual 106 está ilustrada no dispositivo de computação 100, apesar de que alternativamente múltiplas máquinas virtuais podem executar no dispositivo de computação 100. Uma máquina virtual

refere-se a uma implementação de software de um dispositivo de computação físico (ou outra máquina ou sistema) que pode executar programas análogos a um dispositivo de computação físico. A máquina virtual inclui um ou mais componentes virtuais que são similares aos (mas são implementações de software dos) componentes 104. Um sistema operacional assim como outras aplicações podem executar utilizando os componentes virtuais como se estes estivessem utilizando os componentes 104, incluindo executando em processadores virtuais ou núcleos de processador virtuais, acessando a memória virtual, e assim por diante. O sistema operacional e outras aplicações que executam na máquina virtual 106 não precisam ter nenhum conhecimento, e tipicamente não têm conhecimento, que estes estão executando em uma máquina virtual.

[0017] A máquina virtual 106 inclui um sistema operacional 112, uma ou mais aplicações 114, e uma ou mais componentes virtuais 116. O sistema operacional 112 funciona ou executa sobre um ou mais processadores ou núcleos de processador virtuais incluídos como um ou mais dos componentes 116, e gerencia a execução das aplicações 114.

[0018] O gerenciador de máquina virtual 102 inclui um módulo de controle de máquina virtual (VM) 122 e um módulo de gerenciamento de página 124. O módulo de controle de máquina virtual 122 gerencia o mapeamento dos componentes virtuais 116 para os componentes 104, incluindo a programação de processadores ou núcleos de processador virtuais para executar em processadores ou núcleos de processador físicos. O módulo de gerenciamento de página 124 identifica quais páginas são executáveis no modo de núcleo, e pode opcionalmente executar verificações de integridade de código sobre o código para as páginas de memória a serem executáveis no modo de núcleo como abaixo discutido em mais detalhes. Apesar de ilustrados como

dois módulos separados, deve ser notado que a funcionalidade dos módulos 122 e 124 pode ser combinada em um único módulo (por exemplo, a funcionalidade do módulo de gerenciamento de página 124 pode ser incluído no módulo de controle de VM 122).

[0019] O sistema operacional 112 e o gerenciador de máquina virtual 102 gerenciam o armazenamento e acesso à memória que é composta de múltiplos blocos ou porções que são referidos como páginas de memória (ou simplesmente páginas). A memória pode ser, por exemplo, qualquer tipo de memória endereçável a CPU (Unidade de Processamento Central), tal como uma memória volátil (por exemplo, RAM) ou memória não volátil (por exemplo, memória instantânea). Os diferentes programas podem ser alocados páginas de memória, e estes programas podem ser aplicações 114, programas de sistema operacional 112, ou outros componentes ou módulos.

[0020] O sistema operacional 112 e o gerenciador de máquina virtual 102 podem permitir diferentes tipos de acesso às páginas de memória por um programa, tal como um acesso de leitura, acesso de escrita, e acesso de execução. Se um acesso de leitura (também referido como permissão de leitura) for dado a uma página de memória, então o conteúdo da página de memória é permitido ser lido (por exemplo, para um uma ou mais programas específicos). Se um acesso de escrita (também referido como permissão deescrita) for dado a uma página de memória, então o conteúdo é permitido ser escrito na página de memória (por exemplo, por um ou mais programas específicos). Se um acesso de execução (também referido como permissão de execução) for dado a uma página de memória, código armazenado na (também referido como armazenado sobre a) página de memória é permitido ser executado.

[0021] O sistema operacional 112 e/ou uma entidade mais privilegiada do que o sistema operacional 112 (por exemplo, o gerenciador

de máquina virtual 102) pode determinar se dar uma permissão de execução para uma página de memória com base pelo menos em parte na verificação da integridade de código do código sobre a página de memória. Verificar a integridade de código refere-se a verificar a integridade do código (por exemplo, um binário ou suas porções) com base em uma política de integridade de código. Várias diferentes políticas de integridade de código podem ser utilizadas, e a integridade do código pode assim ser verificada em vários diferentes modos. Se a integridade do código for verificada com base na política de integridade de código, então a integridade de código é verificada e o código é permitido executar. No entanto, se a integridade do código não for verificada com base política de integridade de código, então a integridade de código não é verificada e o código não é permitido executar.

[0022] Em uma ou mais modalidades, a política de integridade de código indica que a integridade de código é verificada com base no código tendo sido assinado utilizando certificados digitais que identificam a origem do código (por exemplo, uma entidade que assinou digitalmente o código) e estabelece uma cadeia de confiança para o código. O código é assinado gerando uma assinatura digital com base no código e uma chave criptográfica. Sem a chave criptográfica (ou uma chave correspondente, tal como chave privada de um par de chaves pública / privada) é computacionalmente muito difícil criar uma assinatura que possa ser verificada utilizando a chave criptográfica. No entanto, qualquer entidade com a chave criptográfica (ou uma chave correspondente, tal como uma chave pública de um par de chaves pública / privada) pode utilizar a chave para verificar a assinatura digital executando um algoritmo de verificação de assinatura digital adequado sobre a chave, a assinatura, e o código que foi assinado. Como a assinatura digital está baseada no código, qualquer mudança no código resultará na assinatura digital não sendo verificada. Assim, o certificado

digital permite uma entidade verificar o código para verificar que o código não foi mudado após o código ter sido digitalmente assinado.

[0023] A entidade que verifica o código (por exemplo, o sistema operacional 112 ou o gerenciador de máquina virtual 102) identifica uma ou mais entidades confiáveis. Estas entidades confiáveis podem ser identificadas em diferentes modos, tal como sendo pré-configurada na entidade de verificação, sendo provida por um administrator do dispositivo de computação 100, ou sendo obtida em outro lugar. Uma cadeia de confiança pode ser estabelecida que identifica a entidade confiável assim como uma ou mais outras entidades. A cadeia de confiança refere-se a uma série de entidades começando com a entidade que digitalmente assinou o código e terminando com uma entidade que é confiada pela entidade que verifica o código. Qualquer número de entidades adicionais pode ser incluído na cadeia, cada entidade verificando que esta confia na entidade anterior na cadeia. Por exemplo, assuma que o código é assinado por uma entidade A que não é confiada pela entidade que verifica o código, mas que a entidade D é confiada pela entidade que verifica o código. A cadeia de confiança pode incluir a entidade A que digitalmente assinou o código, uma entidade B que provê um certificado digital verificando que a entidade B confia na entidade A, e a entidade C provendo um certificado digital verificando que a entidade C confia na entidade B, e a entidade D provendo um certificado digital verificando que a entidade D confia na entidade C.

[0024] Se a cadeia de confiança for verificada e o código não foi modificado, então a verificação de integridade de código tem sucesso - a integridade do código é verificada e o código é permitido executar. No entanto, se a cadeia de confiança não for verificada e/ou o código foi modificado, então a verificação de integridade de código falha - a integridade do código não é verificada e o código não é permitido executar.

[0025] Alternativamente, a integridade do código pode ser verifica-

da em outros modos. Por exemplo, código pode ser gerado pela, ou na direção da entidade que está verificando o código. A política de integridade de código pode indicar que tal código é automaticamente tratado como verificado pela entidade, e a verificação de integridade de código para tal código tem sucesso (a integridade do código é verificada e o código é permitido executar). Como outro exemplo, o código pode ser verificado sendo analisado de acordo com várias outras regras ou critérios indicados pela política de integridade de código. Se a análise do código determinar que a política de integridade de código foi satisfeita, então o código é verificado e a verificação de integridade de código tem sucesso (a integridade do código é verificada e o código é permitido executar). No entanto, se a análise do código determinar que a política de integridade de código não foi satisfeita, então o código não é verificado e a verificação de integridade de código falha (a integridade do código não é verificada e o código não é permitido executar).

[0026] Em uma ou mais modalidades, o código de um programa pode ser armazenado em múltiplas páginas de memória, e a verificação de integridade de código para estas múltiplas páginas é executada como um todo. A verificação de integridade de código para o código do programa é executada, e se a verificação de integridade de código tem sucesso então uma permissão de execução é dada para todas múltiplas páginas de memória nas quais o código está armazenado. No entanto, se a verificação de integridade de código falhar então a permissão de execução não é dada a qualquer uma das múltiplas páginas de memória nas quais o código está armazenado.

[0027] Alternativamente, a verificação de integridade de código para cada uma das múltiplas páginas nas quais o código de um programa está armazenado pode ser executada individualmente, e independentemente da verificação de integridade de código para o código

armazenado em outras das múltiplas páginas. Por exemplo, em resposta a uma tentativa de executar o código em uma das múltiplas páginas de memória, a verificação de integridade de código é executada para pelo menos o código naquela página de memória e a permissão de execução é dada ou não dada para a página de memória com base em se a verificação de integridade de código falhar ou ter sucesso.

[0028] Os um ou mais processadores do dispositivo de computação 100 suportam a execução de código em múltiplos diferentes modos, referidos como modo de núcleo (também referido como modo de núcleo, modo de supervisor, ou modo supervisor) e modo de usuário (também referido como modo usuário). As aplicações 114 tipicamente executam em modo de usuário, e o sistema operacional 112 pode incluir alguns componentes de núcleo que executam em modo de núcleo e outros componentes que executam em modo de usuário. O modo de usuário é menos privilegiado (isto é, mais restrito) do que o modo de núcleo. Os drivers instalados ou de outro modo incluídos no sistema operacional 112 para facilitar a comunicação com componentes virtuais 116 podem executar em modo de usuário ou em modo de núcleo. A utilização de modo de núcleo e modo de usuário provê uma proteção adicional para o código que executa em modo de núcleo, tal como por um processador que executa o código impedindo que o código que executa no modo de usuário de acessar a memória utilizada pelo código que executa no modo de núcleo.

[0029] Apesar de referidos aqui como modo de núcleo e modo de usuário, alternativamente uma ou mais modos adicionais podem ser suportados pelos processadores do dispositivo de computação 100. Em tais situações, os modos aqui discutidos podem ser referidos como modo de núcleo e modo não de núcleo, com os modos outros que o modo de núcleo sendo tratados analogamente ao modo de usuário aqui discutido. Alternativamente, os modos aqui discutidos podem ser

referidos como um modo de usuário e modo não de usuário, com os modos outros que o modo de usuário sendo tratados analogamente ao modo de núcleo aqui discutido.

[0030] A Figura 2 ilustra um sistema exemplar 200 que implementa as técnicas de imposição de integridade de código seletiva facilitada por gerenciador de máquina virtual aqui discutidas. O sistema 200 inclui um sistema operacional 112 que executa em uma máquina virtual 106, e um módulo de verificação de integridade de código 202 que é parte de uma entidade privilegiada mais alta 204. O módulo de verificação de integridade de código 202 pode opcionalmente estar incluído em um módulo de gerenciamento de página 124 que executa em um gerenciador de máquina virtual 102 da Figura 1. O módulo de verificação de integridade de código 202 executa a verificação de integridade de código para um código que um processador virtual deseja executar quando o processador virtual está operando no modo de núcleo, provendo resultados de verificação de código de modo de núcleo 206 que indicam se a verificação de integridade de código para o código tem sucesso ou falha. Se a verificação de integridade de código tiver sucesso então o código pode ser executado no modo de núcleo, e se a verificação de integridade de código falhar então o código não pode ser executado no modo de núcleo.

[0031] A entidade privilegiada mais alta 204 refere-se a uma entidade que é mais privilegiada (menos restringida) do que o sistema operacional 112. A entidade privilegiada mais alta 204 pode ser o gerenciador de máquina virtual 102 da Figura 1. A entidade privilegiada mais alta 204 pode alternativamente ser uma ou mais outras entidades, tal como um processador virtual que executa em um modo seguro que um privilégio mais alto do que o modo de núcleo do sistema operacional 112 (por exemplo, um modo seguro gerenciado pelo gerenciador de máquina virtual 102). A entidade privilegiada mais alta pode

ser implementada como software, firmware, e/ou hardware. Em situações nas quais a entidade privilegiada mais alta 204 é uma entidade outra que o gerenciador de máquina virtual, os resultados de verificação de código de modo de núcleo 206 pode ser providos para o gerenciador de máquina virtual, permitindo que o gerenciador de máquina virtual mantenha um registro que identifica o código a integridade do qual foi verificada e assim pode ser executado no modo de núcleo.

[0032] O sistema operacional 112 inclui um módulo de imposição de política 208 que implementa uma ou mais políticas para determinar se o código que um processador virtual deseja executar quando o processador virtual está operando no modo de usuário pode ser executado. Várias diferentes políticas podem ser implementadas pelo módulo de imposição de política 208, tal como executando uma verificação de integridade de código análoga ao módulo de verificação de integridade de código 202, verificando outras características do código ou do sistema operacional 112, e assim por diante. A política implementada pelo módulo de imposição de política 208 pode também ser para executar todo o código no modo de usuário (por exemplo, executar o código no modo de usuário sem executar nenhuma verificação de integridade de código ou outras verificações). O módulo de imposição de política 208 aplica a política para o código quando o processador virtual está operando no modo de usuário, provendo os resultados de avaliação de política 210 indicando se a política é satisfeita (a verificação de política tem sucesso) ou não é satisfeita (a verificação de política falha). Se a verificação de política tiver sucesso então o código pode ser executado no modo de usuário, e se a verificação de política falhar então o código não pode ser executado no modo de usuário.

[0033] Assim, a entidade mais privilegiada 204 executa a verificação de integridade de código (impõe a integridade de código) para páginas no modo de núcleo, enquanto que o sistema operacional 112 em

uma máquina virtual 106 executa verificação de política (impõe a política) para as páginas em modo de usuário. Em uma ou mais modalidades, o módulo de imposição de política 208 é implementado em código de modo de núcleo do sistema operacional 112. Assim, as verificações de política para o código que executa em modo de usuário são feitas pelo código que executa no modo de núcleo, com a integridade de código do código que executa no modo de núcleo tendo sido verificada pelo gerenciador de máquina virtual 102 ou outra entidade mais privilegiada.

[0034] Assim, o gerenciador de máquina virtual 102 é capaz de restringir a execução do código no modo de núcleo para um código que foi verificado pela entidade privilegiada mais alta 204, mas a execução do código em modo de usuário é controlada separadamente pelo sistema operacional 112 que executa na máquina virtual 106. O sistema operacional 112 executa qualquer verificação de código e/ou outras verificações de política com base na configuração do módulo de imposição de política 208, e independentemente da verificação de integridade de código executada pela entidade privilegiada mais alta 204.

[0035] As técnicas aqui discutidas assim proveem um nível adicional de segurança devido ao gerenciador de máquina virtual impedindo que um código comprometido do sistema operacional 112 execute no modo de núcleo. Ao mesmo tempo, no entanto, as técnicas aqui discutidas permitem que a integridade de código do código que executa no modo de usuário seja verificada e/ou outras políticas implementadas pelo sistema operacional 112 conforme apropriado. O sistema operacional 112 ajusta as políticas para execução de código em modo de usuário, permitindo que o sistema operacional 112 suporte situações onde nenhuma verificação de código é executada. Por exemplo, o sistema operacional 112 pode desejar permitir que algum código execute

no modo de usuário sem ser verificado, ou permitir que uma geração de código dinâmica seja executada para algum código que executa no modo de usuário. Tais determinações podem ser feitas pelo sistema operacional 112, todas enquanto o usuário do dispositivo de computação 100 é assegurado que o sistema operacional 112 não foi comprometido porque a integridade de código do sistema operacional 112 foi verificada pelo gerenciador de máquina virtual 102 (ou outra entidade privilegiada mais alta).

[0036] A Figura 3 é um fluxograma que ilustra um processo exemplar 300 para implementar a imposição de integridade de código seletivo facilitada por gerenciador de máquina virtual de acordo com uma ou mais modalidades. O processo 300 é executado pelo menos em parte por um gerenciador de máquina virtual, tal como o gerenciador de máquina virtual 102 das Figuras 1 e 2, e pode ser implementado em software, firmware, hardware, ou suas combinações. O processo 300 está mostrado com um conjunto de atos e não está limitado à ordem mostrada para executar as operações dos vários atos. O processo 300 é um processo exemplar para implementar a imposição de integridade de código seletivo facilitada por gerenciador de máquina virtual; discussões adicionais da implementação da imposição de integridade de código seletivo facilitada por gerenciador de máquina virtual estão aqui incluídas com referência a diferentes figuras.

[0037] No processo 300, uma página de memória que inclui um código executável a ser executado por um processador virtual está identificada (ato 302). A página de memória pode ser identificada diferentes tempos e em resposta a diferentes eventos, tal como uma solicitação por uma máquina virtual ou sistema operacional gerenciado por uma máquina virtual para tornar uma página de memória executável, uma solicitação por uma máquina virtual ou sistema operacional gerenciado por uma máquina virtual para permitir que um programa

seja executável pelo processador virtual, e assim por diante.

[0038] Uma determinação é feita quanto a se a página de memória deve ser executável em um modo de núcleo (ato 304). Esta determinação pode ser feita em vários modos, tal como sendo identificada pela máquina virtual ou sistema operacional quando fazendo a solicitação para tornar a página de memória executável.

[0039] Responsivo a determinar que a página de memória deve ser executável em um modo de núcleo, uma verificação de integridade de código do código executável é feita por uma entidade privilegiada mais alta (ato 306). A entidade privilegiada mais alta é uma entidade mais privilegiada do que um sistema operacional gerenciado por uma máquina virtual como acima discutido. A entidade mais privilegiada pode ser um gerenciador de máquina virtual, ou outra entidade como acima discutido. A verificação de integridade de código é feita verificando a integridade do código com base em uma política de integridade de código em vários modos como acima discutido, por exemplo, verificando o código com base em uma cadeia de confiança sendo verificada e verificação de assinatura digital que o código não foi modificado. A execução do código executável na página de memória é permitida somente se a verificação de integridade de código verificar o código executável (ato 308). A execução do código executável na página de memória pode ser permitida, por exemplo, pelo gerenciador de máquina virtual dando permissão de execução para a página de memória identificada no ato 302.

[0040] Retornando ao ato 304, responsivo a determinar que a página de memória não deve ser executável em um modo de núcleo, um sistema operacional de uma máquina virtual gerenciada pelo gerenciador de máquina virtual é permitido determinar se permitir a execução do código executável com base na política do sistema operacional (ato 310). O sistema operacional pode implementar várias diferentes políti-

cas para determinar se executar o código, incluindo uma verificação de integridade de código, como acima discutido.

[0041] Uma solicitação para tornar uma página de memória executável no ato 302 pode ser direcionada para o gerenciador de máquina virtual, ou alternativamente para a entidade privilegiada mais alta (por exemplo, a entidade privilegiada mais alta 204 da Figura 2). Em uma ou mais modalidades, se o gerenciador de máquina virtual for a entidade privilegiada mais alta, então a solicitação é direcionada para o gerenciador de máquina virtual, e o gerenciador de máquina virtual executa a verificação de integridade de código no ato 306. No entanto, se outra entidade for a entidade privilegiada mais alta, então a solicitação é direcionada para entidade privilegiada mais alta (por exemplo, diretamente ou através do gerenciador de máquina virtual), a entidade privilegiada mais alta executa a verificação de integridade de código no ato 306, e se a verificação de integridade de código no ato 306 verificar o código executável então a entidade privilegiada mais alta notifica o gerenciador de máquina virtual para tornar a página de memória executável no modo de núcleo. O gerenciador de máquina virtual pode tornar a página de memória executável em modo de núcleo, por exemplo, atualizando uma tabela de tradução de endereço de segundo nível como abaixo discutido em mais detalhes.

[0042] Assim, o gerenciador de máquina virtual facilita a imposição de integridade de código seletiva. A integridade de código do modo de núcleo código é imposta pelo gerenciador de máquina virtual (ou outra entidade privilegiada mais alta que alavanca o gerenciador de máquina virtual), enquanto que a imposição de integridade de código de código de modo de usuário é deixada para o sistema operacional que executa na máquina virtual gerenciada pelo gerenciador de máquina virtual.

[0043] Retornando à Figura 1, em uma ou mais modalidades o dispositivo de computação 100 emprega uma memória virtual. A me-

mória virtual refere-se a um espaço de endereço que é mapeado para outro espaço de endereço (por exemplo, memória física). Uma aplicação é atribuída um espaço de memória virtual no qual o código de aplicação é executado e dados são armazenados. Um gerenciador de memória (por exemplo, de um processador) gerencia o mapeamento dos endereços de memória virtual no espaço de memória virtual para endereços no outro espaço de memória. Quando mapeando os endereços de memória virtual de um virtual espaço de endereço de memória para outro espaço de memória, uma tradução de endereço é executada. Uma tabela de tradução de endereço é utilizada para executar este mapeamento, e pode ser alavancada para implementar as técnicas aqui discutidas.

[0044] Em uma ou mais modalidades, um tabela de tradução de endereço está implementada em hardware, tal como por um processador físico que é um componente 104 do dispositivo de computação 100. A tabela de tradução de endereço permite que o gerenciador de máquina virtual 102 seletivamente imponha a integridade de código como abaixo discutido em mais detalhes. Alternativamente, o hardware do dispositivo de computação 100 (por exemplo, um processador físico que é um componente 104) pode utilizar várias outras tabelas, listas, registros, estruturas, e assim por diante para permitir que o gerenciador de máquina virtual 102 seletivamente imponha a integridade de código. Alternativamente, várias outras tabelas, listas, registros, estruturas, e assim por diante implementados em software (por exemplo, como parte do gerenciador de máquina virtual 102 ou como parte de outro componente ou módulo do dispositivo de computação 100) podem ser utilizados para permitir que o gerenciador de máquina virtual 102 seletivamente imponha a integridade de código.

[0045] A Figura 4 ilustra um sistema exemplar 400 que implementa a imposição de integridade de código seletivo facilitada por gerencia-

dor de máquina virtual de acordo com uma ou mais modalidades. O sistema 400 pode ser, por exemplo, o dispositivo de computação 100 da Figura 1. O sistema 400 inclui um processador físico 402, um espaço de memória física 404, um processador virtual 406, e um programa 408. O processador físico 402 pode ser um componente 104 da Figura 1, o espaço de memória física 404 pode ser um componente 104 da Figura 1, o processador virtual 406 pode ser um componente virtual 116 da Figura 1, e o programa 408 pode ser uma aplicação 114 ou parte do sistema operacional 112 da Figura 1. O processador físico 402 inclui um gerenciador de memória 410 que gerencia o acesso ao espaço de memória física 404. O espaço de memória física 404 pode ser várias memórias volátil e/ou não volátil, tal como uma RAM, Memória instantânea, e assim por diante.

[0046] O processador físico 402 atribui um espaço de memória de máquina virtual 412 para o processador virtual 406, e mantém uma tabela de tradução de endereço de segundo nível 414. A tabela de tradução de endereço de segundo nível 414 mapeia os endereços no espaço de memória de máquina virtual 412 para endereços no espaço de memória física 404. Qual endereço do espaço de memória física 404 um endereço específico no espaço de memória de máquina virtual 412 mapeia em qualquer dado tempo pode mudar, e é controlado pelo gerenciador de memória 410. O gerenciador de memória 410 pode mudar mapeamentos, permitindo que múltiplos diferentes processadores virtuais compartilhem o espaço de memória física 404 e/ou permitindo que o espaço de memória de máquina virtual 412 seja maior do que o espaço de memória física 404, utilizando qualquer uma de uma variedade de técnicas públicas e/ou de propriedade.

[0047] O processador virtual 406 inclui um gerenciador de memória 416 que gerenciar o acesso ao espaço de memória de máquina virtual 412. O processador virtual 406 atribui um espaço de memória de

programa 418 para o programa 408, e mantém uma tabela de tradução de endereço de primeiro nível 420. A tabela de tradução de endereço de primeiro nível 420 mapeia os endereços no espaço de memória de programa 418 para endereços no espaço de memória de máquina virtual 412. Qual endereço do espaço de memória de máquina virtual 412 um endereço específico no espaço de memória de programa 418 mapeia em qualquer dado tempo pode mudar, e é controlado pelo gerenciador de memória 416. O gerenciador de memória 416 pode mudar mapeamentos, permitindo que múltiplos diferentes programas compartilhem o espaço de memória de máquina virtual 412 e/ou permitindo que o espaço de memória de programa 418 seja maior do que o espaço de memória de máquina virtual 412, utilizando qualquer uma de uma variedade de técnicas públicas e/ou de propriedade.

[0048] Em resposta a um acesso a um endereço no espaço de memória de programa 418, o gerenciador de memória 416 utiliza a tabela de tradução de endereço de primeiro nível 420 para transladar o endereço de memória no espaço de memória de programa 418 para um endereço no espaço de memória de máquina virtual 412. O acesso pode tomar várias diferentes formas, tal como o endereço de código do programa 408 ser executado por um sistema operacional, o endereço onde os dados a serem lidos está armazenado no programa 408, o endereço onde os dados a serem escritos pelo programa 408 deve ser armazenado, e assim por diante.

[0049] Similarmente, em resposta a um acesso a um endereço no espaço de memória de máquina virtual 412, o gerenciador de memória 410 utiliza a tabela de tradução de endereço de segundo nível 414 para transladar o endereço de memória no espaço de memória de máquina virtual 412 para um endereço no espaço de memória física 404. O acesso pode tomar várias diferentes formas, tal como o endereço de código de um núcleo de sistema operacional (por exemplo, núcleo)

gerenciado pelo processador virtual 406 a ser executado, um endereço do programa 408 a ser executado por um sistema operacional, o endereço onde dados a serem lidos está armazenado pelo programa 408 ou um núcleo de sistema operacional, o endereço onde os dados a serem escritos pelo programa 408 ou um núcleo de sistema operacional deve ser armazenado, e assim por diante.

[0050] Em uma ou mais modalidades, a tabela de tradução de endereço de primeiro nível 420 e a tabela de tradução de endereço de segundo nível 414 mapeiam páginas de endereços ao invés de endereços individuais. Por exemplo, a tabela de tradução de endereço de primeiro nível 420 mapeia as páginas do espaço de memória de programa 418 para páginas do espaço de memória de máquina virtual 412, e a tabela de tradução de endereço de segundo nível 414 mapeia as páginas do espaço de memória de máquina virtual 412 para páginas do espaço de memória física 404. Alternativamente, a tabela 414 e/ou a tabela 420 podem mapear endereços com base em outros grupamentos, tal como individualmente ou outras coleções de subpáginas de endereços, em coleções de múltiplas páginas, e assim por diante.

[0051] O processador virtual 406 pode executar no modo de núcleo ou no modo de usuário. O gerenciador de memória 416 pode manter um registro de páginas de memória que podem executadas em modo de núcleo. Este registro pode estar incluído como parte da tabela de tradução de endereço de primeiro nível 420 ou alternativamente mantido em outros modos. O gerenciador de memória 410 mantém um registro de páginas de memória no espaço de memória de máquina virtual 412 que podem ser executadas no modo de núcleo. Este registro pode estar incluído como parte da tabela de tradução de endereço de segundo nível 414 ou alternativamente mantido em outros modos.

[0052] Quando um acesso é feito para executar um código no processador virtual 406, o endereço do código a ser executado é transla-

dado pela tabela de tradução de endereço de primeiro nível 420 para obter um endereço transladado no espaço de memória de máquina virtual 412, e o endereço transladado é então transladado pela tabela de tradução de endereço de segundo nível 414 para obter um endereço no espaço de memória física 404. Se o acesso for feito enquanto o processador virtual 406 está operando no modo de usuário, então um sistema operacional que executa no processador virtual 406 aplica a política apropriada na determinação se o código pode ser executado.

[0053] No entanto, se o acesso for feito enquanto o processador virtual está operando no modo de núcleo, então uma verificação é feita quanto a se a permissão de execução para o modo de núcleo foi dada para a página. Um registro de se a permissão de execução para o modo de núcleo já foi dada para a página pode opcionalmente ser mantido na tabela de tradução de endereço de segundo nível 414, ou alternativamente em outro local pelo processador físico 402. Em tais situações, se a permissão de execução para o modo de núcleo já foi dada para a página, então uma indicação que a permissão de execução para o modo de núcleo foi dada para a página pode ser retornada para o processador virtual 406 e o sistema operacional que executa no processador virtual 406 permite que o código execute enquanto o processador virtual 406 está no modo de núcleo.

[0054] Se a permissão de execução para o modo de núcleo ainda não foi dada para a página, então o sistema operacional que executa no processador virtual 406 pode opcionalmente solicitar que a página seja tornada executável para o modo de núcleo. O processador virtual 406 passa a solicitação para o gerenciador de máquina virtual (ou outra entidade privilegiada mais alta), a qual executa uma verificação de integridade de código como acima discutido. Se a verificação de integridade de código tiver sucesso então a página é dada permissão de execução para o modo de núcleo, e se a verificação de integridade de

código falhar então a página não é dada permissão de execução para o modo de núcleo.

[0055] Em uma ou mais modalidades, um atributo é incluído na tabela de tradução de endereço de segundo nível 414 para permitir a permissão de execução para o modo de núcleo seja especificada separadamente da permissão de execução ou política para o modo de usuário. Este atributo pode ser codificado na tabela de tradução de endereço de segundo nível 414 em uma variedade de diferentes modos, mas inclui as seguintes três características. Primeiro, o atributo é implementado na tabela de tradução de endereço de segundo nível de modo que o atributo pode ser controlado independentemente para cada tradução de segundo nível (cada tradução executada pelo gerenciador de memória 410 utilizando a tabela de tradução de endereço de segundo nível 414). Segundo, o atributo aplica a permissão de execução de modo de núcleo - o atributo não precisa efetuar (mas alternativamente poderia efetuar) permissão de leitura e/ou escrita no modo de núcleo (e no modo de usuário). Terceiro, o atributo permite que a execução de modo de núcleo seja desabilitada independentemente da execução de modo de usuário, permitindo pelo menos as seguintes duas combinações de execução: 1) não permitir a execução no modo de núcleo, mas permitir a execução no modo de usuário; 2) não permitir a execução no modo de núcleo e invalidade a execução no modo de usuário.

[0056] O pelo menos um atributo pode ser incluído na tabela de tradução de endereço de segundo nível 414 em uma variedade de diferentes modos. Por exemplo, cada entrada na tabela de tradução de endereço de segundo nível pode incluir dois bits, um bit correspondendo à execução de modo de usuário e um bit correspondendo à execução de modo de núcleo. Ao bit que corresponde às páginas de modo de usuário pode ser atribuído um valor (por exemplo, atribuído

um valor de "1", também referido como o bit sendo determinado) para indicar a permissão para executar o código na página em modo de usuário é dada, e atribuído outro valor (por exemplo, atribuído um valor de "0", também referido como o bit sendo apagado) para indicar que a permissão para executar o código na página no modo de usuário não é dada. Similarmente, ao bit que corresponde à execução de modo de núcleo pode ser atribuído um valor (por exemplo, atribuído um valor de "1", também referido como o bit sendo determinado) para indicar que a permissão para executar o código na página em modo de núcleo é dada, e atribuído outro valor (por exemplo, atribuído um valor de "0", também referido como o bit sendo apagado) para indicar que a permissão para executar o código na página no modo de núcleo não é dada.

[0057] Em uma ou mais modalidades, na tabela de tradução de endereço de segundo nível 414, o valor inicial ou padrão para as páginas de memória é não permitir a execução (a permissão de execução não é dada) para o modo de núcleo, mas permitir a execução (a permissão de execução é dada) para o modo de usuário. Se a integridade de código do código em uma página de memória foi verificada pelo gerenciador de máquina virtual (ou outra entidade mais privilegiada), então os valores da página de memória podem ser mudados para permitir a execução (a permissão de execução é dada) para o modo de núcleo para a página assim como para o modo de usuário. No entanto, se integridade de código do código em uma página de memória falhar em uma verificação de integridade de código pelo sistema operacional, o sistema operacional pode opcionalmente mudar (ou solicitar que o gerenciador de máquina virtual mude) os valores para a página de memória para não permitir a execução (a permissão de execução não é dada) para o modo de usuário para a página.

[0058] Deve ser notado que a tabela de tradução de endereço de

segundo nível 414 está sob o controle de um gerenciador de máquina virtual, tal como o gerenciador de máquina virtual 102 da Figura 1. A tabela de tradução de endereço de segundo nível 414 não é acessível pelo software que executa na máquina virtual (por exemplo, as aplicações 114 ou o sistema operacional 112 da máquina virtual 106 da Figura 1). O sistema operacional na máquina virtual tem um controle direto sobre a sua tabela de tradução de endereço de primeiro nível, e o gerenciador de máquina virtual tem exclusivo controle sobre a tabela de tradução de endereço de segundo nível. Assim, tendo um controle de execução de modo de núcleo na tabela de tradução de endereço de segundo nível, o gerenciador de máquina virtual pode eficientemente e facilmente utilizar este controle para conceder ou negar o acesso de execução de modo de núcleo independentemente do gerenciamento do sistema operacional da tabela de tradução de endereço de primeiro nível. Tendo a capacidade de restringir a execução de código de modo de núcleo na tabela de tradução de endereço de segundo nível, o gerenciador de máquina virtual pode impor as suas políticas de integridade de código sem diretamente envolver-se nas atualizações de tabela de tradução de endereço de primeiro nível executadas pelo sistema operacional na máquina virtual.

[0059] Deve ser também notado que apesar de tabelas de tradução de endereço de primeiro e segundo níveis serem acima discutidas, alternativamente a integridade de código imposta por gerenciador de máquina virtual pode ser implementada em software utilizando uma única tabela de tradução de endereço (por exemplo, a tabela de tradução de endereço de primeiro nível). Por exemplo, o gerenciador de máquina virtual pode assumir o controle da tabela de tradução de endereço de primeiro nível (por exemplo, a tabela 420 da Figura 4) e impedir que o sistema operacional de dentro da máquina virtual diretamente acesse a tabela de tradução de endereço de primeiro nível. O

gerenciador de máquina virtual pode então impor as políticas como acima discutido (por exemplo, o gerenciador de máquina virtual pode assegurar que o código executável pelo núcleo é o código que o gerenciador de máquina virtual verificou, mas o gerenciador de máquina virtual não impõe nenhuma restrição sobre a execução de código de modo de usuário).

[0060] Apesar de uma funcionalidade específica ser aqui discutida com referência a módulos específicos, deve ser notado que a funcionalidade de módulos individuais aqui discutida pode ser separada em múltiplos módulos, e/ou pelo menos alguma funcionalidade de múltiplos módulos pode ser combinada em um único módulo. Além disso, um módulo específico aqui discutido como executando uma ação inclui aquele próprio módulo específico executando a ação, ou alternativamente este módulo específico invocando ou de outro modo acessando outro componente ou módulo que executa a ação (ou executa a ação em conjunto com aquele módulo específico). Assim, um módulo específico que executa uma ação inclui aquele próprio módulo específico executando a ação e/ou outro módulo invocado ou de outro modo acessado por aquele módulo específico executando a ação.

[0061] A Figura 5 ilustra um sistema exemplar geralmente em 500 que inclui um dispositivo de computação exemplar 502 que é representativo de um ou mais sistemas e/ou dispositivos que podem implementar as várias técnicas aqui descritas. O dispositivo de computação 502 pode ser, por exemplo, um servidor de um provedor de serviço, um dispositivo associado com um cliente (por exemplo, um dispositivo de cliente), um sistema em chip, e/ou qualquer outro dispositivo de computação ou sistema de computação adequado.

[0062] O dispositivo de computação exemplar 502 como ilustrado inclui um sistema de processamento 504, um ou mais meios legíveis por computador 506, e uma ou mais interfaces I/O 508 que estão co-

municativamente acoplados, uns nos outros. Apesar de não mostrado, o dispositivo de computação 502 pode ainda incluir um barramento de sistema ou outro sistema de transferência de dados e comando que acopla os vários componentes, uns nos outros. Um barramento de sistema pode incluir qualquer uma ou combinação de diferentes estruturas de barramento, tal como um barramento de memória ou controlador de memória, um barramento periférico, um barramento serial universal, e/ou a processador ou barramento local que utiliza qualquer uma de uma variedade de arquiteturas de barramento. Uma variedade de outros exemplos está também contemplada tal como linhas de controle e dados.

[0063] O sistema de processamento 504 é representativo de uma funcionalidade para executar uma ou mais operações utilizando hardware. Consequentemente, o sistema de processamento 504 está ilustrado como incluindo elementos de hardware 510 que podem ser configurados como processadores, blocos funcionais, e assim por diante. Isto pode incluir uma implementação em hardware como um circuito integrado de aplicação específica ou outro dispositivo lógico formado utilizando uma ou mais semicondutores. Os elementos de hardware 510 não estão limitados pelos materiais dos quais estão são formados ou os mecanismos de processamento aqui empregados. Por exemplo, os processadores podem ser compreendidos de semicondutor(es) e/ou transistores (por exemplo, circuito integrado eletrônico (ICs)). Em tal contexto, as instruções executáveis por processador podem ser instruções eletronicamente executáveis.

[0064] Os meios legíveis por computador 506 estão ilustrados como incluindo uma memória / armazenamento 512. A memória / armazenamento 512 representa uma capacidade de memória / armazenamento associada com um ou mais meios legíveis por computador. A memória / armazenamento 512 pode incluir um meio volátil (tal como

memória de acesso randômico (RAM)) e/ou meio não volátil (tal como memória somente memória (ROM), memória instantânea, discos óticos, discos magnéticos, e assim por diante). A memória / armazenamento 512 pode incluir um meio fixo (por exemplo, RAM, ROM, uma unidade rígida fixa, e assim por diante) assim como um meio removível (por exemplo, memória instantânea, uma unidade rígida removível, um disco ótico, e assim por diante). Os meios legíveis por computador 506 podem estar configurados em uma variedade de outros modos como adicionalmente abaixo descrito.

[0065] A(s) interface(s) de entrada / saída 508 são representativas de uma funcionalidade para permitir um usuário inserir comandos e informações em um dispositivo de computação 502, e também permitir que informações sejam apresentadas para o usuário e/ou outros componentes ou dispositivos utilizando vários dispositivos de entrada / saída. Exemplos de dispositivos de entrada incluem um teclado, um dispositivo de controle de cursor (por exemplo, um mouse), um microfone (por exemplo, para entrada de voz), um scanner, uma funcionalidade de toque (por exemplo, sensores capacitivos ou outros que estão configurados para detectar um toque físico), uma câmera (por exemplo, a qual pode empregar comprimentos de onda visíveis ou não visíveis tal como frequências de infravermelho para detectar o movimento que não envolve toque como gestos), e assim por diante. Exemplos de dispositivos de saída incluem um dispositivo de display (por exemplo, um monitor ou projetor), alto-falantes, uma impressora, uma placa de rede, um dispositivo de resposta táctil, e assim por diante. Assim, o dispositivo de computação 502 pode estar configurado em uma variedade de modos como adicionalmente abaixo descrito para suportar a interação de usuário.

[0066] O dispositivo de computação 502 também inclui um gerenciador de máquina virtual 514 (também referido como um hipervisor).

O gerenciador de máquina virtual 514 permite que uma máquina virtual execute no dispositivo de computação 502. O gerenciador de máquina virtual 514 pode ser, por exemplo, um gerenciador de máquina virtual 102 da Figura 1 ou Figura 2.

[0067] Várias técnicas podem ser aqui descritas no contexto geral de software, elementos de hardware, ou módulos de programa. Geralmente, tais módulos incluem rotinas, programas, objetos, elementos, componentes, estruturas de dados, e assim por diante que executam tarefas específicas ou implementam tipos de dados abstratos específicos. Os termos "módulo", "funcionalidade", e "componente" como aqui utilizados geralmente representam software, firmware, hardware, ou uma sua combinação. As características das técnicas aqui descritas são independentes de plataforma, significando que as técnicas podem ser implementadas em uma variedade de plataformas de computação tendo uma variedade de processadores.

[0068] Uma implementação dos módulos e técnicas descritos pode ser armazenada em ou transmitida através de alguma forma de meios legíveis por computador. Os meios legíveis por computador podem incluir uma variedade de meios que podem ser acessados pelo dispositivo de computação 502. Como exemplo, e não limitação, os meios legíveis por computador podem incluir "meios de armazenamento legíveis por computador" e "meios de sinal legíveis por computador".

[0069] "Meios de armazenamento legíveis por computador" referem-se a meios e/ou dispositivos que permitem um armazenamento persistente de informações e/ou armazenamento que é tangível, em contraste com a mera transmissão de sinal, ondas portadoras, ou sinais por si. Assim, os meios de armazenamento legíveis por computador referem-se a meios que não contêm sinal. Os meios de armazenamento legíveis por computador incluem um hardware tal como meios voláteis e não voláteis, removíveis e não removíveis e/ou dispositi-

vos de armazenamento implementados em um método ou tecnologia adequado para armazenamento de informações tal como instruções legíveis por computador, estruturas de dados, módulos de programa, elementos / circuitos lógicos, ou outros dados. Exemplos de meios de armazenamento legíveis por computador podem incluir, mas não estão limitados a, RAM, ROM, EEPROM, memória instantânea ou outra tecnologia de memória, CD-ROM, discos versáteis digitais (DVD) ou outro armazenamento ótico, discos rígidos, cassetes magnéticos, fita magnética, armazenamento de disco magnético ou outros dispositivos de armazenamento magnéticos, ou outros dispositivos de armazenamento magnético, ou outro dispositivo de armazenamento, meios tangíveis, ou artigo de manufatura adequado para armazenar as informações desejáveis e o qual pode ser acessado por um computador.

[0070] "Meios de sinal legíveis por computador" referem-se a um meio que contém sinal que está configurado para transmitir instruções para o hardware do dispositivo de computação 502, tal como através de uma rede. Os meios de sinal tipicamente podem incorporar instruções legíveis por computador, estruturas de dados, módulos de programa, ou outros dados em um sinal de dados modulado, tal como ondas portadoras, sinais de dados, ou outro mecanismo de transporte. Os meios de sinal também incluem quaisquer meios de fornecimento de informações. O termo "sinal de dados modulado" significa um sinal que tem uma ou mais de suas características ajustadas ou mudadas de tal modo a codificar as informações no sinal. Como exemplo, e não limitação, os meios de comunicação incluem meios com fio tal como uma rede com fio ou conexão com fio direto, e meios sem fio tal como acústico, RF, infravermelho, e outros meios sem fio.

[0071] Como anteriormente descrito, os elementos de hardware 510 e os meios legíveis por computador 506 são representativos de instruções, módulos, lógica de dispositivo programável e/ou lógica de

dispositivo fixa implementada em uma forma de hardware que pode ser empregado em algumas modalidades para implementar pelo menos alguns aspectos das técnicas aqui descritas. Os elementos de hardware podem incluir componentes de um circuito integrado ou um sistema em chip, um circuito integrado de aplicação específica (ASIC), uma rede de portas programáveis no campo (FPGA), um dispositivo lógico programável complexo (CPLD), e outras implementações em silício ou outros dispositivos de hardware. Neste contexto, um elemento de hardware pode operar como um dispositivo de processamento que executa tarefas de programa definidas por instruções, módulos, e/ou lógica incorporados pelo elemento de hardware assim como um dispositivo de hardware utilizado para armazenar instruções para execução, por exemplo, pelos meios de armazenamento legíveis por computador anteriormente descrito.

[0072] Combinações dos acima podem também ser empregadas para implementar várias técnicas e módulos aqui descritos. Consequentemente, software, hardware, ou módulos de programa e outros módulos de programa podem ser implementados como uma ou mais instruções e/ou lógica incorporada em alguma forma de meios de armazenamento legíveis por computador e/ou por um ou mais elementos de hardware 510. O dispositivo de computação 502 pode estar configurado para implementar instruções e/ou funções específicas que correspondem aos módulos de software e/ou hardware. Consequentemente, a implementação de módulos como um módulo que é executável pelo dispositivo de computação 502 como software pode ser conseguida pelo menos parcialmente em hardware, por exemplo, através da utilização de meios de armazenamento legíveis por computador e/ou elementos de hardware 510 do sistema de processamento. As instruções e/ou funções podem executáveis / operáveis por um ou mais artigos de manufatura (por exemplo, uma ou mais dispositivos de

computação 502 e/ou sistemas de processamento 504) para implementar as técnicas, módulos, e exemplos aqui descritos.

[0073] Como ainda ilustrado na Figura 5, o sistema exemplar 500 permite ambientes ubíquos para uma experiência de usuário uniforme quando executando as aplicações em um computador pessoal (PC), um dispositivo de televisão, e/ou um dispositivo móvel. Os serviços e aplicações executam substancialmente similares em todos os três ambientes para uma experiência de usuário comum quando transicionando de um dispositivo para o seguinte enquanto utilizando uma aplicação, reproduzindo um videogame, assistindo um vídeo, e assim por diante.

[0074] No sistema exemplar 500, múltiplos dispositivos estão interconectados através de um dispositivo de computação central. O dispositivo de computação central pode ser local aos múltiplos dispositivos ou pode estar localizado remotamente dos múltiplos dispositivos. Em uma ou mais modalidades, o dispositivo de computação central pode ser uma nuvem de um ou mais computadores servidores que estão conectados aos múltiplos dispositivos através de uma rede, a Internet, ou outra conexão de comunicação de dados.

[0075] Em uma ou mais modalidades, esta arquitetura de interconexão permite que a funcionalidade seja fornecida através de múltiplos dispositivos para prover uma experiência comum e uniforme para um usuário dos múltiplos dispositivos. Cada um dos múltiplos dispositivos pode ter diferentes requisitos e capacidades físicas, e o dispositivo de computação central utiliza uma plataforma para permitir o fornecimento de uma experiência para o dispositivo que é tanto modelada para o dispositivo e quanto ainda comum para todos os dispositivos. Em uma ou mais modalidades, uma classe de dispositivos alvo é criada e as experiências podem ser modeladas para a classe genérica de dispositivos. Uma classe de dispositivos pode ser definida por características,

físicas, tipos de utilização, ou outras características comuns dos dispositivos.

[0076] Em várias implementações, o dispositivo de computação 502 pode assumir uma variedade de diferentes configurações, tal como para usos de computador 516, móvel 518, e televisão 520. Cada uma destas configurações inclui dispositivos que podem ter construções e capacidades geralmente diferentes, e assim o dispositivo de computação 502 pode ser configurado de acordo com uma ou mais das diferentes classes de dispositivo. Por exemplo, o dispositivo de computação 502 pode ser implementado como a classe de computador 516 de um dispositivo que inclui um computador pessoal, computador desktop, um computador de múltiplas telas, computador laptop, netbook, e assim por diante.

[0077] O dispositivo de computação 502 pode também ser implementado como a classe de dispositivo móvel 518 que inclui dispositivos móveis, tal como um telefone móvel, um reprodutor de música portátil, um dispositivo de jogos portátil, um computador tablet, um computador de múltiplas telas, e assim por diante. O dispositivo de computação 502 pode também ser implementado como a classe de dispositivo de televisão 520 que inclui dispositivos possuindo ou conectados a telas geralmente maiores em ambientes de visualização casual. Estes dispositivos incluem televisões, decodificadores, consoles de jogos, e assim por diante.

[0078] As técnicas aqui descritas podem ser suportadas por estas várias configurações do dispositivo de computação 502 e não estão limitadas aos exemplos específicos das técnicas aqui descritas. Esta funcionalidade pode também ser implementada toda ou em parte através da utilização de um sistema distribuído, tal como sobre uma "nuvem" 522 através de uma plataforma 524 como abaixo descrito.

[0079] A nuvem 522 inclui e/ou é representativa de uma platafor-

ma 524 para recursos 526. A plataforma 524 abstrai a funcionalidade subjacente de hardware (por exemplo, servidores) e recursos de software da nuvem 522. Os recursos 526 podem incluir aplicações e/ou dados que podem ser utilizados enquanto o processamento de computador é executado em servidores que estão remotos do dispositivo de computação 502. Os recursos 526 podem também incluir serviços providos sobre a Internet e/ou através de uma rede de assinante, tal como uma rede de celular ou Wi-Fi.

[0080] A plataforma 524 pode abstrair recursos e funções para conectar o dispositivo de computação 502 com outros dispositivos de computação. A plataforma 524 pode também servir para abstrair a escalagem de recursos para prover um nível de escala correspondente para a demanda encontrada para os recursos 526 que são implementados através da plataforma 524. Consequentemente, uma modalidade de dispositivo interconectado, a implementação de funcionalidade aqui descrita pode distribuída através de todo o sistema 500. Por exemplo, a funcionalidade pode ser implementada em parte no dispositivo de computação 502 assim como através da plataforma 524 que abstrai a funcionalidade da nuvem 522.

[0081] Apesar do assunto ter sido descrito em uma linguagem específica para características estruturas e/ou atos metodológicos, deve ser compreendido que o assunto definido nas concretizações anexas não está necessariamente limitado às características ou atos específicos acima descritos. Ao invés as características ou atos específicos acima descritos estão descritos como forma exemplares de implementar as concretizações.

REIVINDICAÇÕES

1. Método implementado em um dispositivo de computação (100, 502), o método **caracterizado pelo fato de que** compreende as etapas de:

identificar (302), em resposta a uma solicitação por uma máquina virtual (106) ou sistema operacional (112) executado na máquina virtual (106) para fazer uma página de memória executável, a página de memória, a página de memória compreendendo uma dentre uma pluralidade de páginas de memória armazenando código para um programa incluindo código executável para ser executado por um processador virtual de uma a máquina virtual (106), a máquina virtual (106) sendo gerenciada por um gerenciador de máquina virtual (102, 204, 514);

determinar (304), em reposta à solicitação para fazer uma página de memória executável, se a página de memória identificada da pluralidade de páginas de memória armazenando código para o programa deve ser executável em um modo de núcleo ou um modo de usuário;

executar (306), por uma entidade mais privilegiada (204, 102) do que o um sistema operacional (112) da máquina virtual (106) e em resposta a determinar que a página de memória identificada da pluralidade de páginas de memória armazenando código para o programa deve ser executável no modo de núcleo, uma verificação de integridade de código do código executável armazenado na pluralidade de páginas de memória e permitir (308) execução do código executável somente se a verificação de integridade de código verificar o código executável armazenado na pluralidade de páginas de memória; e

permitir (310), em resposta a determinar que a página de memória identificada da pluralidade de páginas de memória armazenando código para o programa não deve ser executável no modo de núcleo, o sistema operacional (112) da máquina virtual (106) a deter-

minar se permite execução do código executável de todas dentre a pluralidade de páginas de memória armazenando código para o programa no modo de usuário.

2. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** a entidade mais privilegiada (204, 102) do que o sistema operacional (112) da máquina virtual (106) compreende o gerenciador de máquina virtual (102, 204, 514), a determinação se a página de memória identificada deve ser executada em modo de núcleo sendo executada pelo gerenciador de máquina virtual (102, 204, 514), e a permissão ao sistema operacional (112) da máquina virtual (106) para determinar se permite execução do código executável em um modo de usuário sendo executada pelo gerenciador de máquina virtual (102, 204, 514).

3. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** ainda compreende executar, pelo sistema operacional (112) e em resposta a determinar que a página de memória identificada não deve ser executável no modo de núcleo, a verificação de integridade de código do código executável e permitir execução do código executável somente se a verificação de integridade de código pelo sistema operacional (112) verifica o código executável.

4. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** permitir execução do código executável compreende configurar um atributo de permissão de execução de uma página de memória na qual o código está armazenado, a configuração do atributo de permissão de execução compreendendo configurar o atributo de permissão de execução para permitir execução do código executável para modo de núcleo.

5. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** a execução da verificação de integridade de código compreende verificar uma cadeia de confiança para o código executá-

vel, e usar uma assinatura digital para verificar que o código executável não foi alterado desde que foi assinado digitalmente.

6. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** o processador virtual inclui uma tabela de tradução de endereço de primeiro nível, o dispositivo de computação (100, 502) ainda incluindo um processador físico incluindo uma tabela de tradução de endereço de segundo nível, um atributo da tabela de tradução de endereço de segundo nível permitindo permissão de execução para modo de núcleo do processador virtual a ser especificada separadamente de permissão de execução ou política para modo de usuário do processador virtual.

7. Método, de acordo com a reivindicação 1, **caracterizado pelo fato de que** o dispositivo de computação (100, 502) ainda compreende um componente incluindo um atributo permitindo permissão de execução para modo de núcleo do processador virtual para ser especificada separadamente de permissão de execução ou política para modo de usuário do processador virtual.

8. Método, de acordo com a reivindicação 7, **caracterizado pelo fato de que** o atributo permite a permissão de execução para modo de núcleo do processador virtual a ser especificada separadamente de permissão de execução ou política para modo de usuário do processador virtual em uma base por página de memória.

9. Método, de acordo com a reivindicação 7, **caracterizado pelo fato de que** o atributo se aplica a permissão de execução de modo de núcleo, e o atributo não afeta permissão de leitura em modo de núcleo nem permissão de escrita em modo de núcleo.

10.Método, de acordo com a reivindicação 7, **caracterizado pelo fato de que** o atributo permite uma combinação de execução que não permite execução de código na página de memória identificada pelo processador virtual no modo de núcleo mas permite execução do

código na página de memória identificada pelo processador virtual no modo de usuário, e uma combinação adicional de execução que não permite execução do código na página de memória identificada pelo processador virtual em modo de núcleo e não permite execução do código na página de memória identificada pelo processador virtual no modo de usuário.

11. Dispositivo de computação (100, 502), **caracterizado pelo fato de que** inclui um sistema operacional (112), um gerenciador de máquina virtual (102, 204, 514), e um processador, o processador sendo programado com instruções para:

determinar (302), em reposta a uma solicitação por uma máquina virtual (106) para fazer uma página de memória de múltiplas páginas de memória armazenando código para um programa executável e para cada página de memória das múltiplas páginas de memória armazenando código para o programa, se a página de memória deve ser executável em um modo de núcleo ou um modo de usuário;

permitir (308), se a página de memória deve ser executável no modo de núcleo, o gerenciador de máquina virtual (102, 204, 514) a restringir execução de modo de núcleo de todas dentre as múltiplas páginas de memória armazenando código para o programa para permitir execução do código somente se a integridade das múltiplas páginas de memória armazenando código para o programa foi verificada por uma entidade privilegiada mais alta que é mais privilegiada do que o sistema operacional (112); e

permitir (310), se a página de memória deve ser executável no modo de usuário, execução de modo de usuário de todas dentre as múltiplas páginas de memória armazenando código para o programa sem levar em consideração se integridade de código nas páginas de memória armazenando código para o programa foi verificada pela entidade privilegiada mais alta.

12. Dispositivo de computação (100, 502), de acordo com a reivindicação 11, **caracterizado pelo fato de que** o sistema operacional (112) é ainda configurado para permitir execução de modo de usuário de todas dentre as múltiplas páginas de memória armazenando código para o programa se a integridade das múltiplas páginas de memória foi verificada pelo sistema operacional (112).

13. Dispositivo de computação (100, 502), de acordo com a reivindicação 11, **caracterizado pelo fato de que** o processador é ainda programado com instruções para configurar um atributo de permissão de execução de uma página de memória armazenando código, a integridade da qual foi verificada pelo gerenciador de máquina virtual (102, 204, 514).

14. Dispositivo de computação (100, 502), de acordo com a reivindicação 11, **caracterizado pelo fato de que** o processador inclui uma tabela de tradução de endereço, um atributo da qual permite permissão de execução para execução de modo de núcleo de páginas de memória a serem especificadas separadamente de permissão de execução para execução de modo de usuário de páginas de memória.

15. Dispositivo de computação (100, 502), de acordo com a reivindicação 11, **caracterizado pelo fato de que** o processador ainda inclui um componente de hardware tendo um atributo permitindo permissão de execução para execução de modo de núcleo de páginas de memória a serem especificadas separadamente de política de execução para execução de modo de usuário de páginas de memória.

16. Dispositivo de computação (100, 502), de acordo com a reivindicação 15, **caracterizado pelo fato de que** o atributo permite permissão de execução para execução de modo de núcleo de páginas de memória a serem especificadas separadamente de política de execução para execução de modo de usuário de páginas de memória em uma base por página de memória.

17. Dispositivo de computação (100, 502) de acordo com a reivindicação 15, **caracterizado pelo fato de que** o atributo se aplica a permissão de execução de modo de núcleo para páginas de memória, e o atributo não afeta permissão de leitura em modo de núcleo nem permissão de escrita em modo de núcleo para páginas de memória.

18. Dispositivo de computação (100, 502) de acordo com a reivindicação 15, **caracterizado pelo fato de que** o atributo permite uma combinação de execução que não permite execução de modo de núcleo de código na página de memória mas permite execução de modo de usuário do código na página de memória, e uma combinação adicional que não permite execução de modo de núcleo do código na página de memória e nem execução de modo de usuário do código na página de memória.

19. Meio de armazenamento legível por computador tendo um método, **caracterizado pelo fato de que** o método compreende:

identificar (302), em resposta a uma solicitação por uma máquina virtual (106) ou sistema operacional (112) executado na máquina virtual (106) para fazer uma página de memória executável, a página de memória, a página de memória compreende uma dentre uma pluralidade de páginas de memória armazenando código para um programa incluindo código executável para ser executado por um processador virtual da máquina virtual (106) de um dispositivo de computação (100, 502), a máquina virtual (106) sendo gerenciada por um gerenciador de máquina virtual (102, 204, 514) do dispositivo de computação (100, 502);

determinar (304), em reposta à solicitação para fazer uma página de memória executável, se a página de memória identificada da pluralidade de páginas de memória armazenando código para o programa deve ser executável em um modo de núcleo ou um modo de usuário;

em resposta a determinar que a página de memória da pluralidade de páginas de memória armazenando código para o programa deve ser executável no modo de núcleo:

executar (306), pelo gerenciador de máquina virtual (102, 204, 514), uma verificação de integridade de código do código executável da página de memória identificada da pluralidade de páginas de memória armazenando código para o programa, e

permitir (308) execução do código executável de todas dentre a pluralidade de páginas de memória armazenando código para o programa para o modo de núcleo somente se a verificação de integridade de código verificar o código executável da página de memória identificada; e

em resposta a determinar que a página de memória da pluralidade de páginas de memória armazenando código para o programa não deve ser executável no modo de núcleo, permitir (310) um sistema operacional (112) da máquina virtual (106) a determinar se deve permitir execução do código executável de todas dentre a pluralidade de páginas de memória armazenando código para o programa no modo de usuário, o sistema operacional (112) permitindo execução do código executável somente se a política do sistema operacional (112) é satisfeita.

20. Meio de armazenamento legível por computador, de acordo com a reivindicação 19, **caracterizado pelo fato de que** o dispositivo de computação (100, 502) ainda compreende um componente incluindo um atributo permitindo permissão de execução para modo de núcleo do processador virtual a ser especificada separadamente de permissão de execução ou política para modo de usuário do processador virtual.

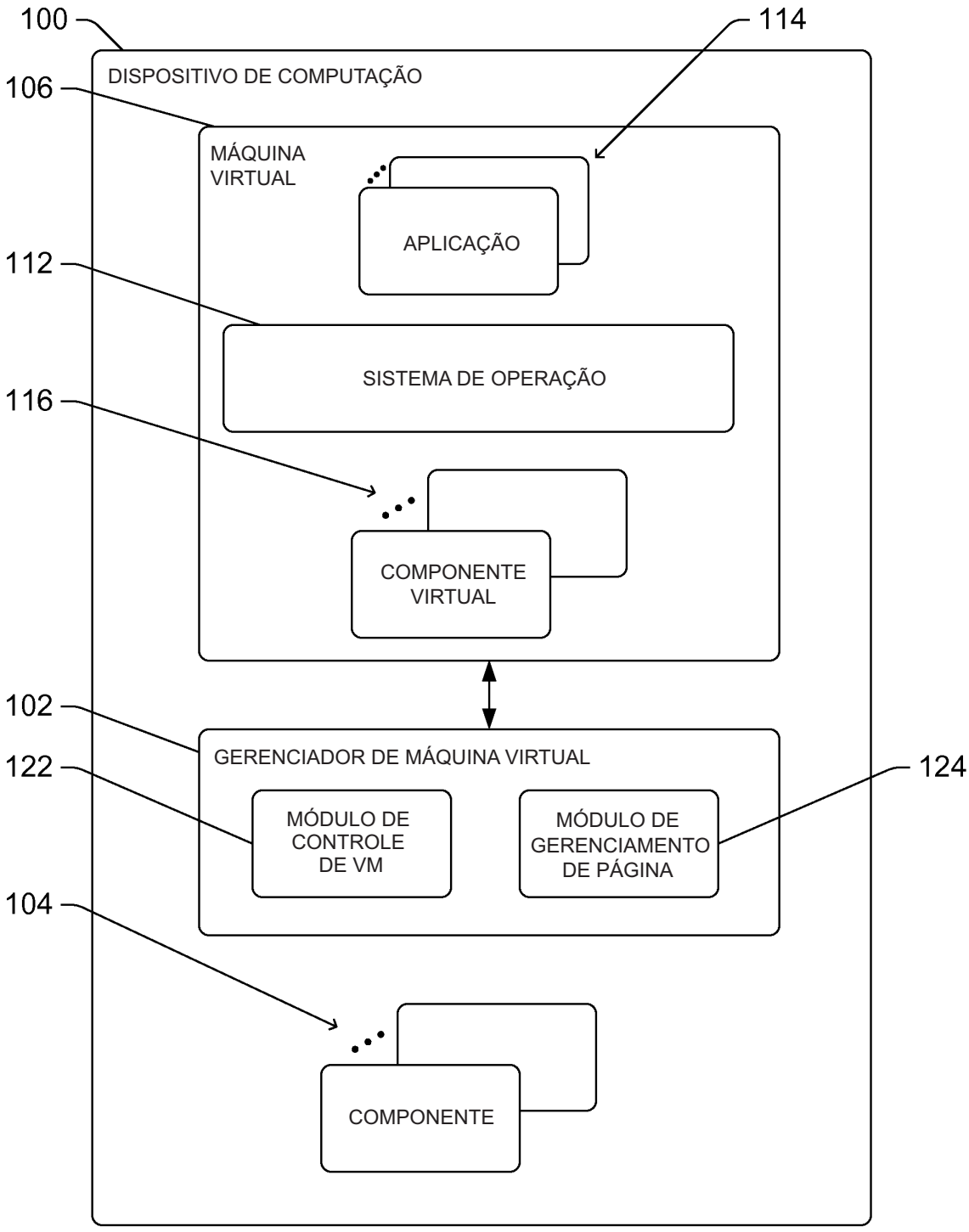
100
114
DISPOSITIVO DE COMPUTAÇÃO
106
MÁQUINA VIRTUAL
APLICAÇÃO
112
SISTEMA DE OPERAÇÃO
116
COMPONENTE VIRTUAL
102
GERENCIADOR DE MÁQUINA VIRTUAL
122
MÓDULO DE CONTROLE DE VM
MÓDULO DE GERENCIAMENTO DE PÁGINA
124
104
COMPONENTE

Fig. 1

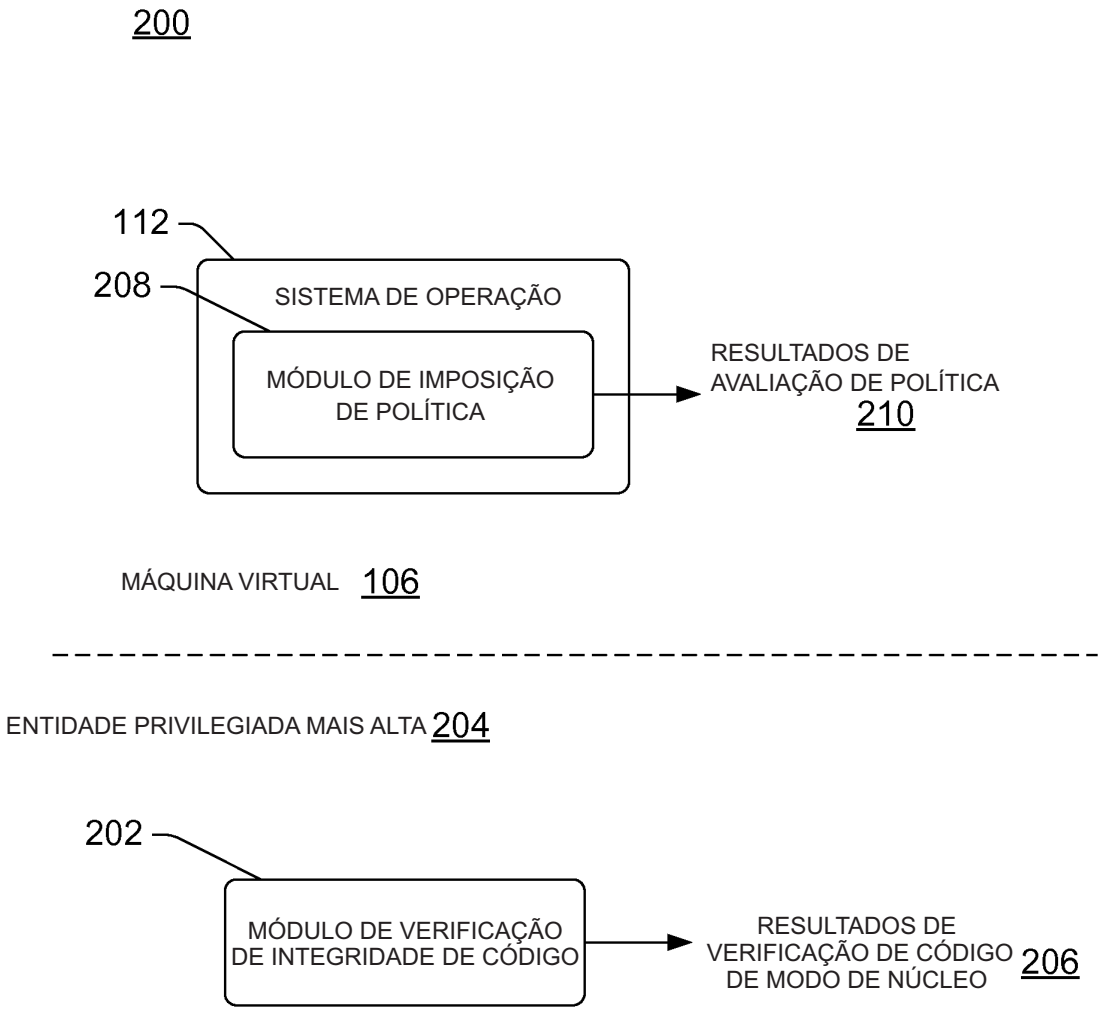
200
112
208
SISTEMA DE OPERAÇÃO
MÓDULO DE IMPOSIÇÃO DE POLÍTICA
RESULTADOS DE AVALIAÇÃO DE POLÍTICA
210
MÁQUINA VIRTUAL 106
ENTIDADE PRIVILEGIADA MAIS ALTA 204
202
MÓDULO DE VERIFICAÇÃO DE INTEGRIDADE DE CÓDIGO
RESULTADOS DE VERIFICAÇÃO DE CÓDIGO DE MODO DE NÚCLEO
206

Fig. 2

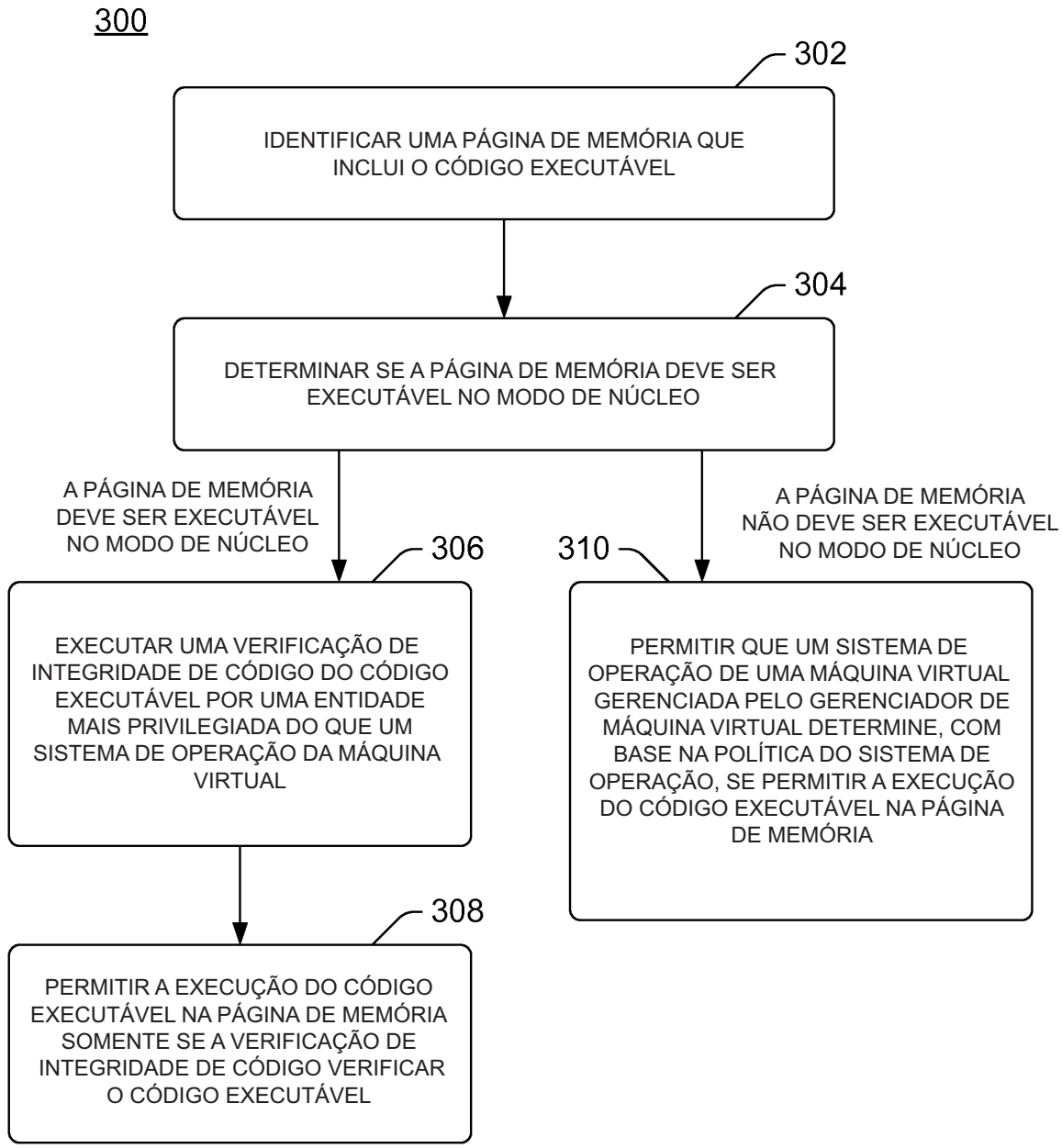
300
302
IDENTIFICAR UMA PÁGINA DE MEMÓRIA QUE INCLUI O CÓDIGO EXECUTÁVEL
304
DETERMINAR SE A PÁGINA DE MEMÓRIA DEVE SER EXECUTÁVEL NO MODO DE NÚCLEO
A PÁGINA DE MEMÓRIA DEVE SER EXECUTÁVEL NO MODO DE NÚCLEO
306
EXECUTAR UMA VERIFICAÇÃO DE INTEGRIDADE DE CÓDIGO DO CÓDIGO EXECUTÁVEL POR UMA ENTIDADE MAIS PRIVILEGIADA DO QUE UM SISTEMA DE OPERAÇÃO DA MÁQUINA VIRTUAL
308
PERMITIR A EXECUÇÃO DO CÓDIGO EXECUTÁVEL NA PÁGINA DE MEMÓRIA SOMENTE SE A VERIFICAÇÃO DE INTEGRIDADE DE CÓDIGO VERIFICAR O CÓDIGO EXECUTÁVEL
A PÁGINA DE MEMÓRIA NÃO DEVE SER EXECUTÁVEL NO MODO DE NÚCLEO
310
PERMITIR QUE UM SISTEMA DE OPERAÇÃO DE UMA MÁQUINA VIRTUAL GERENCIADA PELO GERENCIADOR DE MÁQUINA VIRTUAL DETERMINE, COM BASE NA POLÍTICA DO SISTEMA DE OPERAÇÃO, SE PERMITIR A EXECUÇÃO DO CÓDIGO EXECUTÁVEL NA PÁGINA DE MEMÓRIA

Fig. 3

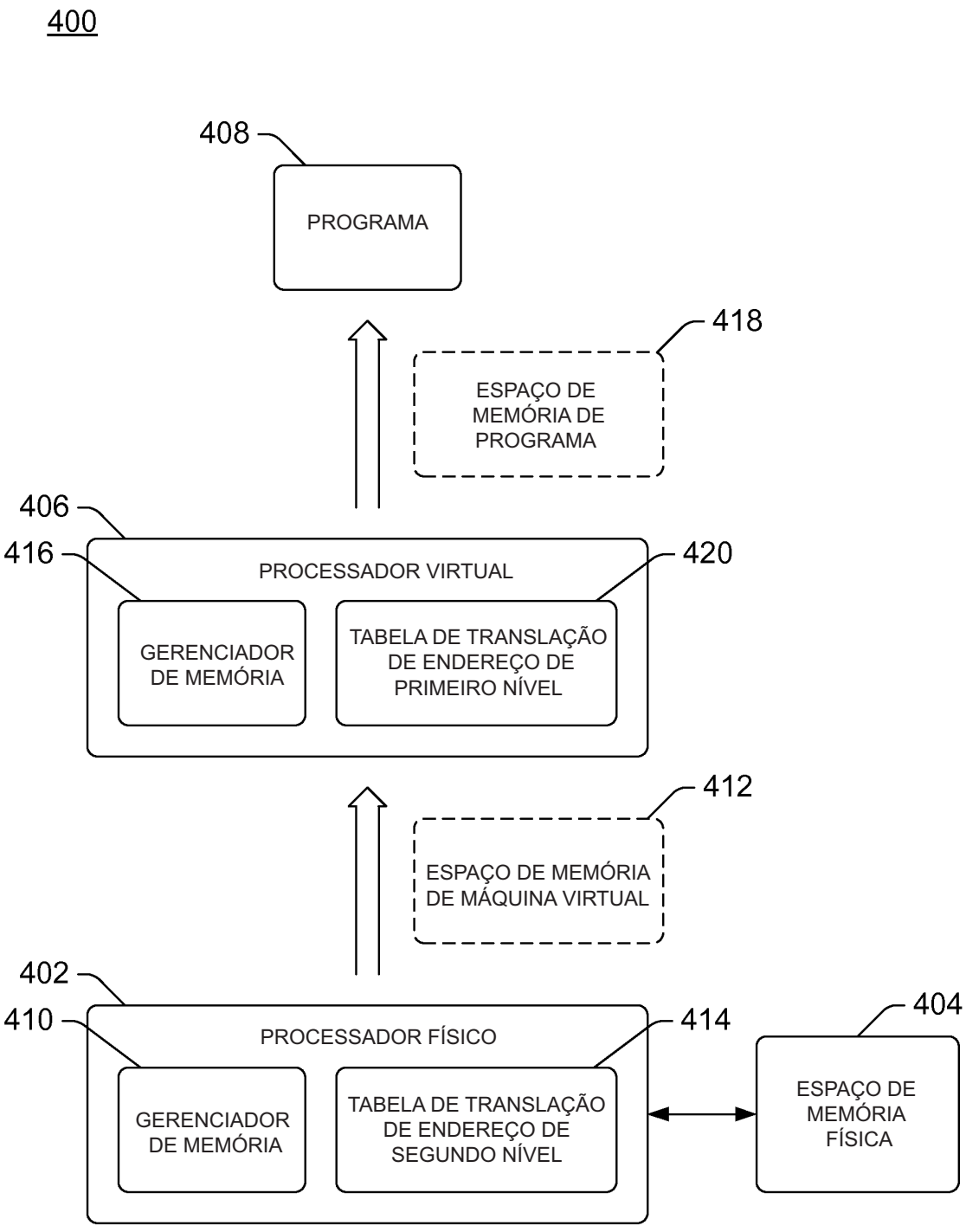
400
408
PROGRAMA
418
ESPAÇO DE MEMÓRIA DE PROGRAMA
406
416
PROCESSADOR VIRTUAL
420
GERENCIADOR DE MEMÓRIA
TABELA DE TRANSLAÇÃO DE ENDEREÇO DE PRIMEIRO NÍVEL
412
ESPAÇO DE MEMÓRIA DE MÁQUINA VIRTUAL
402
410
PROCESSADOR FÍSICO
414
404
GERENCIADOR DE MEMÓRIA
TABELA DE TRANSLAÇÃO DE ENDEREÇO DE SEGUNDO NÍVEL
ESPAÇO DE MEMÓRIA FÍSICA

Fig. 4

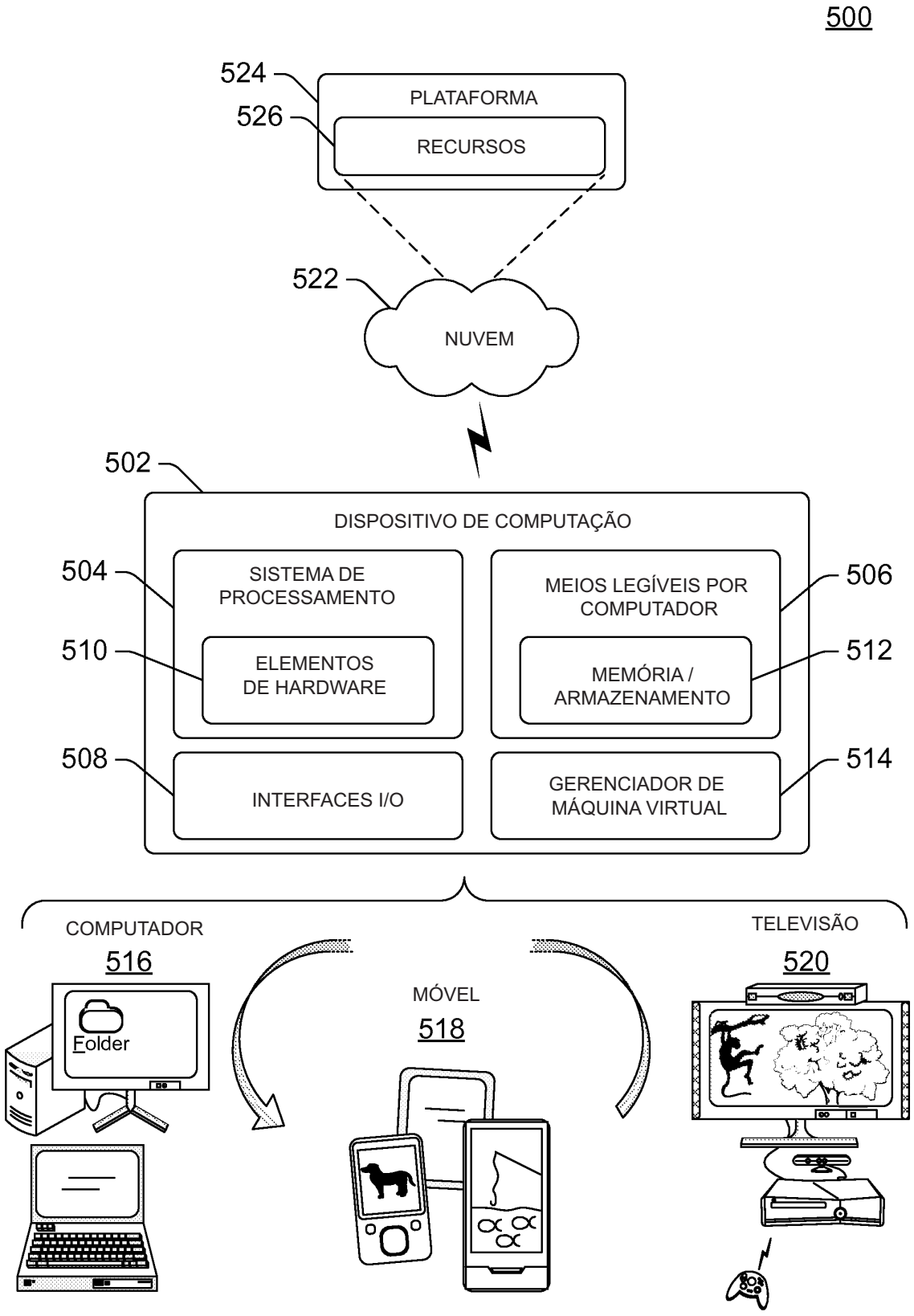
500
524
PLATAFORMA
526
RECURSOS
522
NUVEM
502
DISPOSITIVO DE COMPUTAÇÃO
504
SISTEMA DE PROCESSAMENTO
510
ELEMENTOS DE HARDWARE
506
MEIOS LEGÍVEIS POR COMPUTADOR
512
MEMÓRIA / ARMAZENAMENTO
508
INTERFACES I/O
514
GERENCIADOR DE MÁQUINA VIRTUAL
COMPUTADOR
516
Folder
MÓVEL
518
TELEVISÃO
520

Fig. 5