(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第6397500号
(P6397500)

(45)発行日　平成30年9月26日(2018.9.26)　(24)登録日　平成30年9月7日(2018.9.7)

(51)Int.Cl.　　F I
G06F 21/51 (2013.01)　　G06F 21/51
G06F 21/64 (2013.01)　　G06F 21/64

請求項の数 10　(全 21 頁)

(21)出願番号　特願2016-543935(P2016-543935)
(86)(22)出願日　平成26年9月12日(2014.9.12)
(65)公表番号　特表2016-535373(P2016-535373A)
(43)公表日　平成28年11月10日(2016.11.10)
(86)国際出願番号　PCT/US2014/055290
(87)国際公開番号　WO2015/041930
(87)国際公開日　平成27年3月26日(2015.3.26)
審査請求日　平成29年9月12日(2017.9.12)
(31)優先権主張番号　61/879,068
(32)優先日　平成25年9月17日(2013.9.17)
(33)優先権主張国　米国(US)
(31)優先権主張番号　14/179,378
(32)優先日　平成26年2月12日(2014.2.12)
(33)優先権主張国　米国(US)

(73)特許権者　314015767
マイクロソフト　テクノロジー　ライセンシング，エルエルシー
アメリカ合衆国　ワシントン州　９８０５２　レッドモンド　ワン　マイクロソフト　ウェイ
(74)代理人　100140109
弁理士　小野　新次郎
(74)代理人　100075270
弁理士　小林　泰
(74)代理人　100101373
弁理士　竹内　茂雄
(74)代理人　100118902
弁理士　山本　修

最終頁に続く

(54)【発明の名称】仮想マシン・マネージャーによって支援される選択的コード完全性強制

(57)【特許請求の範囲】
【請求項１】
計算デバイスにおいて実装される方法であって、
仮想マシンまたは該仮想マシンで実行されているオペレーティング・システムによる、メモリー・ページを実行可能にする要求に応答して、前記メモリー・ページを識別するステップであって、前記メモリー・ページが、前記仮想マシンの仮想プロセッサーにより実行されるべき実行可能コードを含むプログラムのためのコードを格納した複数のメモリー・ページのうちの１つを含み、前記仮想マシンが仮想マシン・マネージャーによって管理される、ステップと、
前記メモリー・ページを識別したことに応答して、前記プログラムのためのコードを格納した前記複数のメモリー・ページのうちの識別した前記メモリー・ページがカーネル・モードであるいはユーザー・モードで実行可能にすべきか否か判定するステップと、
前記仮想マシンの前記オペレーティング・システムよりも多くの特権が与えられたエンティティによって、前記プログラムのためのコードを格納した前記複数のメモリー・ページのうちの識別した前記メモリー・ページが前記カーネル・モードで実行可能にすべきであると判定したことに応答して、前記複数のメモリー・ページに格納された前記実行可能コードのコード完全性チェックを実行し、前記コード完全性チェックが前記複数のメモリー・ページに格納された前記実行可能コードを確認した場合にのみ前記実行可能コードの実行を許すステップと、
前記プログラムのためのコードを格納した前記複数のメモリー・ページのうちの識別し

た前記メモリー・ページが前記カーネル・モードで実行可能にするべきでないと判定したことに応答して、前記仮想マシンの前記オペレーティング・システムに、前記プログラムのためのコードを格納した前記複数のメモリー・ページのすべてのメモリー・ページの前記実行可能コードの前記ユーザー・モードでの実行を許すか否か判定させるステップと、を含む、方法。

【請求項２】

請求項１記載の方法において、前記仮想マシンのオペレーティング・システムよりも多くの特権が与えられた前記エンティティが、仮想マシン・マネージャーを含み、識別した前記メモリー・ページがカーネル・モードで実行可能にすべきか否か判定するステップは、前記仮想マシン・マネージャーで実行され、前記仮想マシンの前記オペレーティング・システムに前記実行可能コードの前記ユーザー・モードでの実行を許すか否か判定させるステップは、前記仮想マシン・マネージャーで実行される、方法。

【請求項３】

請求項１記載の方法であって、更に、前記オペレーティング・システムによって、識別した前記メモリー・ページが前記カーネル・モードで実行可能にすべきでないと判定したことに応答して、前記実行可能コードのコード完全性チェックを実行し、前記オペレーティング・システムによるコード完全性チェックが前記実行可能コードを確認した場合にのみ、前記実行可能コードの実行を許すステップを含む、方法。

【請求項４】

請求項１記載の方法において、前記計算デバイスが、更に、前記仮想プロセッサーのユーザー・モードに対する実行許可またはポリシーとは別に、前記仮想プロセッサーのカーネル・モードに対する実行許可を指定させる属性を含むコンポーネントを含む、方法。

【請求項５】

オペレーティング・システムと、仮想マシン・マネージャーと、プロセッサーとを含む計算デバイスであって、前記プロセッサーが、

仮想マシンによる、プログラムのためのコードを格納した複数のメモリー・ページのうちの１つのメモリー・ページを実行可能にする要求に応答して、前記プログラムのためのコードを格納した前記複数のメモリー・ページのうちの各メモリー・ページに関し、前記メモリー・ページがカーネル・モードであるいはユーザー・モードで実行可能にすべきか否か判定し、

前記メモリー・ページが前記カーネル・モードで実行可能にすべき場合に、前記プログラムのためのコードを格納した前記複数のメモリー・ページのすべてのメモリー・ページのカーネル・モード実行を、前記仮想マシン・マネージャーに制限させて、前記プログラムのためのコードを格納した前記複数のメモリー・ページの完全性が、前記オペレーティング・システムよりも多くの特権が与えられた上位特権エンティティによって確認されている場合にのみ、前記コードの実行を許し、

前記メモリー・ページが前記ユーザー・モードで実行可能にすべき場合に、前記プログラムのためのコードを格納した前記複数のメモリー・ページ上のコードの完全性が前記上位特権エンティティによって確認されているか否かには関係なく、前記プログラムのためのコードを格納した前記複数のメモリー・ページのすべてのメモリー・ページのユーザー・モード実行を許す、

命令でプログラムされた、計算デバイス。

【請求項６】

請求項５記載の計算デバイスにおいて、前記プロセッサーが、更に、前記仮想マシン・マネージャーによって完全性が確認されているコードを格納するメモリー・ページの実行許可属性を設定する命令でプログラムされた、計算デバイス。

【請求項７】

請求項５記載の計算デバイスにおいて、前記プロセッサーが、アドレス変換表を含み、その属性が、メモリー・ページのカーネル・モード実行に対する実行許可を、メモリー・ページのユーザー・モード実行に対する実行許可とは別に指定させる、計算デバイス。

【請求項８】
請求項５記載の計算デバイスにおいて、前記プロセッサーが、更に、メモリー・ページのカーネル・モード実行に対する実行許可を、メモリー・ページのユーザー・モード実行に対する実行ポリシーとは別に指定させる属性を有するハードウェア・コンポーネントを含む、計算デバイス。
【請求項９】
請求項８記載の計算デバイスにおいて、前記属性が、メモリー・ページのカーネル・モード実行に対する実行許可を、メモリー・ページ毎のメモリー・ページのユーザー・モード実行に対する実行ポリシーとは別に指定させる、計算デバイス。
【請求項１０】
請求項８記載の計算デバイスにおいて、前記属性が、メモリー・ページにおけるコードのカーネル・モード実行を禁止するが前記メモリー・ページにおける前記コードのユーザー・モード実行を許す１つの実行の組み合わせと、前記メモリー・ページにおける前記コードのカーネル・モード実行および前記メモリー・ページにおける前記コードのユーザー・モード実行の双方を禁止する追加の組み合わせとを許す、計算デバイス。
【発明の詳細な説明】
【背景技術】
【０００１】
[0001] 計算技術が進歩するに連れて、計算デバイスは増々相互接続されつつある。この相互接続は多くの便益を与えるが、その問題がない訳ではない。このような問題の１つは、計算デバイスが悪意のプログラムに増々晒されることである。悪意のプログラムは、計算デバイスから情報を盗み、計算デバイスを使用不可にし、計算デバイスを使用して他の計算デバイスに対して攻撃を始める等によって、異なる方法で動作する可能性がある。計算デバイスを悪意のプログラムから保護するために開発された技法もあるが、このような悪意のプログラムは残っており、ユーザーのコンピューターに感染したとき、苛立たしいユーザー体験を招く可能性がある。
【発明の概要】
【０００２】
[0002] この摘要は、詳細な説明において以下で更に説明する概念から選択したものを、簡略化した形態で紹介するために設けられている。この摘要は、特許請求する主題の主要な特徴や必須の特徴を特定することを意図するのではなく、特許請求する主題の範囲を限定するために使用されることを意図するのでもない。
【０００３】
[0003] １つ以上の形態によれば、仮想マシンの仮想プロセッサーによって実行される実行可能コードを含むメモリー・ページが識別され、仮想マシンは仮想マシン・マネージャーによって管理される。メモリー・ページがカーネル・モードで実行可能にするべきか否かについて判定を行う。メモリー・ページがカーネル・モードで実行可能にするべきであると判定したことに応答して、実行可能コードのコード完全性チェックが実行され、コード完全性チェックが実行可能コードを確認した場合にのみ、実行可能コードの実行が、カーネル・モードに対して許される。メモリー・ページがカーネル・モードで実行可能にするべきでないと判定したことに応答して、仮想マシンのオペレーティング・システムは、実行可能コードの実行を許すか否か判定することができる。
【０００４】
[0004] １つ以上の形態によれば、計算デバイスは、オペレーティング・システム、仮想マシン・マネージャー、およびプロセッサーを含む。プロセッサーは、仮想マシン・マネージャーに、メモリー・ページのカーネル・モード実行を、上位特権エンティティによって完全性が確認されたコードを有するメモリー・ページに制限させるように構成され、上位の特権エンティティはオペレーティング・システムよりも多くの特権が与えられる。しかし、プロセッサーは、メモリー・ページにおけるコードの完全性が上位特権エンティティによって確認されたか否かには関係なく、メモリー・ページのユーザー・モード実行

を許すように構成される。
【図面の簡単な説明】
【０００５】
[0005] 同様の特徴を指すために、図面にわたって同じ番号を使用する。
【図１】図１は、１つ以上の実施形態にしたがって、本明細書において説明する技法を実現する計算デバイス例を示すブロック図である。
【図２】図２は、１つ以上の実施形態にしたがって、本明細書において説明する、仮想マシン・マネージャーによって支援される選択的コード完全性強制技法を実現するシステム例を示す。
【図３】図３は、１つ以上の実施形態にしたがって、仮想マシン・マネージャーによって支援される選択的コード完全性強制を実施するプロセス例を示すフローチャートである。
【図４】図４は、１つ以上の実施形態にしたがって、仮想マシン・マネージャーによって支援される選択的コード完全性強制を実施する他のシステム例を示す。
【図５】図５は、本明細書において説明する種々の技法を実現することができる１つ以上のシステムおよび／またはデバイスを表す計算デバイス例を含むシステム例を示す。
【発明を実施するための形態】
【０００６】
[0011] 本明細書では仮想マシン・マネージャーによって支援される選択的コード完全性強制について説明する。仮想マシンとは、物理デバイスに類似したプログラムを実行することができる物理デバイスのソフトウェア実現例である。仮想マシン、および物理デバイスのハードウェアへのアクセスは、物理デバイスにおける仮想マシン・マネージャーによって管理される。仮想マシンおよび仮想マシン・マネージャー・アクセス・メモリーは、メモリー・ページ（または単にページ）と呼ばれる複数のブロックまたは部分で構成される。コード完全性は、物理デバイスに対する悪意のコードに対抗する保護を促進するために使用される。コード完全性とは、コード完全性ポリシーに基づいて検証される(verify)コード（例えば、バイナリー）の完全性を指す。コード完全性ポリシーに基づいてコードが確認された(verify)場合、コードの完全性が確認されたことになり、コードは実行することが許される。そうでない場合、コードの完全性は確認されず、コードは実行することを許されない。
【０００７】
[0012] プロセッサーは、カーネル・モードまたはユーザー・モードでコードを実行することができる。仮想マシンの仮想プロセッサーがカーネル・モードで実行しているとき、仮想プロセッサーは、仮想マシン・マネージャーによって（または仮想プロセッサーにおいて実行するオペレーティング・システムよりも多くの特権が与えられた他のエンティティによって）完全性が確認されたコードのみを実行する。コードを含むメモリー・ページのコード完全性が確認され、カーネル・モードで実行している仮想プロセッサーは、仮想マシン・マネージャー（または他のもっと特権が与えられたエンティティ）がメモリー・ページにおいてコードのコード完全性を確認した場合にのみ、メモリー・ページにおいてコードを実行することができる。しかしながら、仮想プロセッサーがユーザー・モードで実行しているとき、仮想プロセッサーにおいて実行しているオペレーティング・システムは、コードを実行することができるか否か判定を行う。オペレーティング・システムは、コードをユーザー・モードで実行できるか否か判定するために、種々のポリシーの内任意のもの（例えば、コードの種々の異なるチェックまたは検証の内任意のものを実行する）を適用することができ、任意に、コードのチェックや検証を全く実行せずに、ユーザー・モードでコードを実行することを含む。
【０００８】
[0013] 図１は、１つ以上の実施形態にしたがって、本明細書において説明する技法を実現する計算デバイス例１００を示すブロック図である。計算デバイス１００は、種々の異なるタイプのデバイスの内任意のものにすることができる。例えば、計算デバイス１００は、デスクトップ・コンピューター、サーバー・コンピューター、ラップトップまたは

ネットブック・コンピューター、タブレットまたはノートパッド・コンピューター、移動局、娯楽用アプライアンス、通信可能にディスプレイ・デバイスに結合されたセット・トップ・ボックス、テレビジョンまたは他のディスプレイ・デバイス、セルラーまたは他のワイヤレス電話機、ゲーム・コンソール、自動車用コンピューター、ウェアラブル・コンピューター等とすることができる。

【０００９】

[0014] 計算デバイス１００は、ハイパーバイザーとも呼ばれる仮想マシン・マネージャー１０２、および１つ以上のコンポーネント１０４を含む。仮想マシン・マネージャー１０２は、コンポーネント１０４によって設けられる機能へのアクセスを管理する。あるいは、仮想マシン・マネージャー１０２は、ホスト・オペレーティング・システム（図示せず）においても実行することができ、この場合、ホスト・オペレーティング・システムが、コンポーネント１０４によって設けられる機能へのアクセスを管理する。

【００１０】

[0015] コンポーネント１０４は、種々の異なるプロセッサー・コンポーネント、入力／出力（Ｉ／Ｏ）コンポーネント、および／または他のコンポーネントまたはデバイスとすることができる。例えば、コンポーネント１０４は、１つ以上のプロセッサーまたはプロセッサー・コア、１つ以上のメモリー・コンポーネント（例えば、揮発性および／または不揮発性メモリー）、１つ以上の記憶デバイス（例えば、光および／または磁気ディスク、フラッシュ・メモリー・デバイス）、１つ以上の通信コンポーネント（例えば、有線および／またはワイヤレス・ネットワーク・アダプター）、これらの組み合わせ等を含むことができる。計算デバイス１００の一部として図示するが、コンポーネント１０４の１つ以上（例えば、１つ以上の記憶デバイス）は、計算デバイス１００の外部に実装することができる。仮想マシン・マネージャー１０２を含む、計算デバイス１００において実行する種々のコンポーネントまたはモジュールは、コンポーネント１０４によって設けられるこの機能に直接、および／または他のコンポーネントまたはモジュールを介して間接的にアクセスすることができる。

【００１１】

[0016] 仮想マシン・マネージャー１０２は、仮想マシン１０６が計算デバイス１００において実行することを許す。１つの仮想マシン１０６が計算デバイス１００内に示されているが、代わりに、複数の仮想マシンが計算デバイス１００において実行することもできる。仮想マシンとは、物理計算デバイスと類似するプログラムを実行することができる物理計算デバイス（または他のマシンまたはシステム）のソフトウェア実現例を指す。仮想マシンは、コンポーネント１０４と同様（であるがそのソフトウェア実現例）である１つ以上の仮想コンポーネントを含む。オペレーティング・システムおよび他のアプリケーションは、仮想コンポーネントを使用して、コンポーネント１０４を使用して行うかのように、実行することができ、仮想プロセッサーまたは仮想プロセッサー・コアにおいて実行すること、仮想メモリーにアクセスすること等を含む。仮想マシン１０６において実行するオペレーティング・システムおよび他のアプリケーションは、これらが仮想マシンにおいて実行していることを知る必要はなく、そして通例ではそれを知らない。

【００１２】

[0017] 仮想マシン１０６は、オペレーティング・システム１１２、１つ以上のアプリケーション１１４、および１つ以上の仮想コンポーネント１１６を含む。オペレーティング・システム１１２は、１つ以上の仮想プロセッサー、またはコンポーネント１１６の内１つ以上として含まれるプロセッサー・コアにおいて実行し(run or execute)、アプリケーション１１４の実行を管理する。

【００１３】

[0018] 仮想マシン・マネージャー１０２は、仮想マシン（ＶＭ）制御モジュール１２２、およびページ管理モジュール１２４を含む。仮想マシン制御モジュール１２２は、仮想コンポーネント１１６のコンポーネント１０４に対するマッピングを管理し、物理プロセッサーまたはプロセッサー・コアにおいて実行するための仮想プロセッサーまたはプロ

セッサー・コアのスケジューリングを含む。ページ管理モジュール１２４は、どのページがカーネル・モードで実行可能か識別し、任意に、メモリー・ページがカーネル・モードで実行可能であることに関して、コードに対してコード完全性チェックを実行することができる。これについては、以下で更に詳しく説明する。２つの別のモジュールとして示されているが、モジュール１２２および１２４の機能を１つのモジュールに組み合わせることができる（例えば、ページ管理モジュール１２４の機能をＶＭ制御モジュール１２２に含ませることができる）ことは、注記してしかるべきである。

【００１４】

[0019] オペレーティング・システム１１２および仮想マシン・マネージャー１０２は、メモリー・ページ（または単にページ）と呼ばれる複数のブロックまたは部分で構成されるメモリーの格納およびメモリーへのアクセスを管理する。メモリーは、例えば、揮発性メモリー（例えば、ＲＡＭ）または不揮発性メモリー（例えば、フラッシュ・メモリー）のような、ＣＰＵ（中央処理ユニット）アドレス可能なメモリーの内任意のものとすることができる。異なるプログラムにメモリー・ページを割り当てることができ、これらのプログラムはアプリケーション１１４、オペレーティング・システム１１２のプログラム、あるいは他のコンポーネントまたはモジュールとすることができる。

【００１５】

[0020] オペレーティング・システム１１２および仮想マシン・マネージャー１０２は、リード・アクセス、ライト・アクセス、および実行アクセスというような、プログラムによるメモリー・ページへの異なるタイプのアクセスを許すことができる。リード・アクセス（リード許可とも呼ぶ）がメモリー・ページに与えられた場合、このメモリー・ページのコンテンツを読み出す（例えば、特定の１つ以上のプログラムに）ことが許される。ライト・アクセス（ライト許可とも呼ぶ）がメモリー・ページに与えられた場合、このメモリー・ページにコンテンツを書き込む（例えば、特定の１つ以上のプログラムによって）ことが許される。実行アクセス（実行許可とも呼ぶ）がメモリー・ページに与えられると、このメモリー・ページに(in)格納されている（上(on)に格納されているとも言う）コードを実行することが許される。

【００１６】

[0021] オペレーティング・システム１１２および／またはオペレーティング・システム１１２よりも多くの特権が与えられたエンティティ（例えば、仮想マシン・マネージャー１０２）は、少なくとも部分的に、メモリー・ページ上におけるコードのコード完全性を検証することに基づいて、メモリー・ページに実行許可を与えるか否か判定することができる。コード完全性を検証するとは、コード完全性ポリシーに基づいて、コード（例えば、バイナリーまたはその一部）の完全性を検証することを言う。種々の異なるコード完全性ポリシーを使用することができ、つまり種々の異なるやり方でコードの完全性を検証することができる。コード完全性ポリシーに基づいてコードの完全性が確認された場合、コード完全性が確認され、コードは実行することが許される。しかしながら、コード完全性ポリシーに基づいてコードの完全性が確認されない場合、コード完全性は確認されず、コードは実行することが許されない。

【００１７】

[0022] １つ以上の実施形態では、コード完全性ポリシーは、コードの出所(origin)（例えば、コードにディジタル的に署名したエンティティ）を識別するディジタル証明書を使用してコードが署名されていることに基づいて、コードの完全性が確認され、そのコードに対して信頼チェーンが作られた(establish)ことを示す。コードおよび暗号鍵に基づいてディジタル署名を生成することによって、コードに署名される。暗号鍵（または公開／秘密鍵対の秘密鍵のような対応する鍵）がないと、暗号鍵を使用して検証することができる署名を作成することは、計算的に非常に難しい。しかしながら、暗号鍵（または公開／秘密鍵対の公開鍵のような、対応する鍵）を有する任意のエンティティは、この鍵を使用して、鍵、署名、および署名されたコードに対して適したディジタル署名検証アルゴリズムを実行することによって、ディジタル署名を検証することができる。ディジタル署名

がコードに基づくので、コードに対する変更はいずれも、ディジタル署名が確認されない結果となる。つまり、ディジタル証明書は、コードを検証するエンティティに、コードがディジタル的に署名された後にコードが変更されていないことを確認することを可能にする。

【００１８】

[0023] コードを検証するエンティティ（例えば、オペレーティング・システム１１２または仮想マシン・マネージャー１０２）は、１つ以上の信頼されたエンティティを識別する。これらの信頼されたエンティティは、検証エンティティにおいて予め構成される、計算デバイス１００のアドミニストレーターによって提供される、または任意の場所で得られるというように、異なるやり方で識別することができる。信頼されたエンティティおよび１つ以上の他のエンティティを識別する信頼チェーンを作ることができる。信頼チェーンとは、コードにディジタル的に署名したエンティティから始まり、コードを検証するエンティティによって信頼されたエンティティで終わる一連のエンティティを指す。このチェーンには、任意の数の追加のエンティティを含むことができ、各エンティティが、チェーンにおける直前のエンティティをそれが信頼することを検証する。例えば、コードを検証するエンティティによって信頼されていないエンティティＡによってコードが署名されたが、エンティティＤはコードを検証するエンティティによって信頼されていると仮定する。信頼チェーンは、コードにディジタル的に署名したエンティティＡ、エンティティＢがエンティティＡを信頼することを検証するディジタル証明書を与えるエンティティＢ、エンティティＣがエンティティＢを信頼することを検証するディジタル証明書を与えるエンティティＣ、およびエンティティＤがエンティティＣを信頼することを検証するディジタル証明書を与えるエンティティＤを含むことができる。

【００１９】

[0024] 信頼チェーンが確認され、コードが変更されていない場合、コード完全性チェックは成功する。即ち、コードの完全性が確認され、コードは実行することが許される。しかしながら、信頼チェーンが確認されない場合、および／またはコードが変更されている場合、コード完全性チェックは失敗となる。即ち、コードの完全性は確認されず、コードは実行することが許されない。

【００２０】

[0025] あるいは、コードの完全性を他のやり方で検証することもできる。例えば、コードを検証しているエンティティによって、またはこのエンティティの指令でコードを生成することができる。コード完全性ポリシーは、このようなコードが、エンティティによって確認されると自動的に処理され、そしてこのようなコードに対するコード完全性チェックは成功する（コードの完全性が確認され、コードは実行することを許される）ことを示すことができる。他の例として、コード完全性ポリシーによって示される種々の他の規則または判断基準にしたがって分析することによって、コードを検証することができる。このコードの分析が、コード完全性ポリシーが満たされたことを判定した場合、コードは確認され、コード完全性チェックは成功する（コードの完全性が確認され、コードは実行することが許される）。しかしながら、コードの分析が、コード完全性ポリシーが満たされなかったことを判定した場合、コードは確認されず、コード完全性チェックは失敗となる（コードの完全性は確認されず、コードは実行することが許されない）。

【００２１】

[0026] １つ以上の実施形態では、プログラムのコードは、複数のメモリー・ページに格納することができ、これら複数のページに対するコード完全性チェックが全体として実行される。プログラムのコードに対するコード完全性チェックが実行され、このコード完全性チェックが成功した場合、コードが格納されている複数のメモリー・ページの全てに実行許可が与えられる。しかしながら、コード完全性チェックが失敗した場合、コードが格納されている複数のメモリー・ページのいずれにも実行許可は与えられない。

【００２２】

[0027] あるいは、プログラムのコードが格納されている複数のページの各々に対して

コード完全性チェックを個々に、そして複数のページの他のものに格納されているコードに対するコード完全性チェックとは独立して、実行することもできる。例えば、複数のメモリー・ページの１つにおいてコードを実行しようとする試みに応答して、そのメモリー・ページ上において少なくともこのコードに対してコード完全性チェックを実行し、コード完全性チェックが失敗か成功かに基づいて、実行許可をこのメモリー・ページに与えるか、または与えない。

【００２３】

[0028] 計算デバイス１００の１つ以上のプロセッサーは、複数の異なるモードでのコードの実行をサポートする。これらのモードは、カーネル・モード（カーネル－モード、スーパーバイザー・モード、またはスーパーバイザー－モードとも呼ばれる）およびユーザー・モード（ユーザー－モードとも呼ばれる）と呼ばれる。アプリケーション１１４は、通例、ユーザー・モードで実行し、オペレーティング・システム１１２は、カーネル・モードで実行するいくつかのコア・コンポーネントと、ユーザー・モードで実行する他のコンポーネントとを含むことができる。ユーザー・モードは、カーネル・モードよりも与えられる特権が少ない（即ち、制限が多い）。仮想コンポーネント１１６との通信を容易にするためにオペレーティング・システム１１２にインストールされたまたはそれ以外で含まれるドライバーは、ユーザー・モードまたはカーネル・モードで実行することができる。カーネル・モードおよびユーザー・モードの使用によって、ユーザー・モードで実行するコードが、カーネル・モードで実行するコードによって使用されるメモリーにアクセスすることを禁止するコードをプロセッサーが実行することによってというようにして、カーネル・モードで実行するコードに追加の保護を設ける。

【００２４】

[0029] ここではカーネル・モードおよびユーザー・モードと呼ぶが、代わりに、１つ以上の追加のモードが、計算デバイス１００のプロセッサーによってサポートされてもよい。このような状況では、本明細書において説明するモードをカーネル・モードおよび非カーネル・モードと呼ぶことができ、カーネル・モード以外のモードは、本明細書において説明したユーザー・モードに類似して処理される。あるいは、本明細書において説明したモードをユーザー・モードおよび非ユーザー・モードと呼ぶこともでき、ユーザー・モード以外のモードは、本明細書において説明したカーネル・モードに類似して処理される。

【００２５】

[0030] 図２は、本明細書において説明する、仮想マシン・マネージャーによって支援される選択的コード完全性強制技法を実現するシステム例２００を示す。システム２００は、仮想マシン１０６において実行するオペレーティング・システム１１２、およびより多くの特権が与えられるエンティティ２０４の一部であるコード完全性検証モジュール２０２を含む。コード完全性検証モジュール２０２は、任意に、図１の仮想マシン・マネージャー１０２において実行するページ管理モジュール１２４に含むことができる。コード完全性検証モジュール２０２は、仮想プロセッサーがカーネル・モードで動作しているときにこの仮想プロセッサーが実行することを望むコードに対してコード完全性検証を実行し、コードに対するコード完全性検証が成功か失敗かを示すカーネル・モード・コード検証結果２０６を与える。コード完全性検証が成功した場合、コードをカーネル・モードで実行することができ、コード完全性検証が失敗した場合、コードをカーネル・モードで実行することができない。

【００２６】

[0031] 上位(higher)特権エンティティ２０４とは、オペレーティング・システム１１２よりも多い特権が与えられた（制限が少ない）エンティティを指す。上位特権エンティティ２０４は、図１の仮想マシン・マネージャー１０２とすることができる。代わりに、上位特権エンティティ２０４は、オペレーティング・システム１１２のカーネル・モードよりも上位の特権であるセキュア・モード（例えば、仮想マシン・マネージャー１０２によって管理されるセキュア・モード）で実行する仮想プロセッサーのような、１つ以上の

他のエンティティとすることができる。上位特権エンティティは、ソフトウェア、ファームウェア、および／またはハードウェアとして実現することができる。上位特権エンティティ２０４が仮想マシン・マネージャー以外のエンティティである状況では、カーネル・モード・コード検証結果２０６を仮想マシン・マネージャーに供給することができ、仮想マシン・マネージャーが、完全性が確認されたがってカーネル・モードで実行することができるコードを識別するレコードを維持することを可能にする。

【００２７】

[0032] オペレーティング・システム１１２は、ポリシー強制モジュール２０８を含む。ポリシー強制モジュール２０８は、仮想プロセッサーがユーザー・モードで動作しているときに仮想プロセッサーが実行することを望むコードを実行することができるか否か判定するための１つ以上のポリシーを実施する。種々の異なるポリシーは、コード完全性検証モジュール２０２に類似したコード完全性検証を実行し、コードまたはオペレーティング・システム１１２の他の特性をチェックする等というように、ポリシー強制モジュール２０８によって実施することができる。ポリシー強制モジュール２０８によって実施されるポリシーは、ユーザー・モードで全てのコードを実行するためであることもできる（例えば、コード完全性チェックまたは他の検証を全く実行することなく、ユーザー・モードでコードを実行する）。ポリシー強制モジュール２０８は、仮想プロセッサーがユーザー・モードで動作しているときに、コードに対してポリシーを適用し、ポリシーが満たされるか（ポリシー・チェックが成功する）または満たされないか（ポリシー・チェックが失敗する）を示すポリシー強制結果２１０を供給する。ポリシー・チェックが成功した場合、ユーザー・モードでコードを実行することができ、ポリシー・チェックが失敗した場合、ユーザー・モードでコードを実行することはできない。

【００２８】

[0033] このように、より多くの特権が与えられたエンティティ２０４は、カーネル・モードのページに対するコード完全性検証を実行し（コード完全性を強制する）、一方仮想マシン１０６におけるオペレーティング・システム１１２は、ユーザー・モードのページに対するポリシー・チェックを実行する（ポリシーを強制する）。１つ以上の実施形態では、ポリシー強制モジュール２０８は、オペレーティング・システム１１２のカーネル・モード・コードで実装される。このように、ユーザー・モードで実行するコードに対するポリシー・チェックは、カーネル・モードで実行するコードによって行われ、カーネル・モードで実行するコードのコード完全性は、仮想マシン・マネージャー１０または他のより多くの特権が与えられたエンティティによって確認されている。

【００２９】

[0034] このように、仮想マシン・マネージャー１０２は、カーネル・モードにおけるコードの実行を、上位特権エンティティ２０４によって確認されたコードに制限することができるが、ユーザー・モードにおけるコードの実行は、仮想マシン１０６において実行するオペレーティング・システム１１２によって別に制御される。オペレーティング・システム１１２は、ポリシー強制モジュール２０８の構成に基づいて、そして上位特権エンティティ２０４によって実行されるコード完全性検証とは独立して、任意のコード検証および／または他のポリシー・チェックを実行する。

【００３０】

[0035] つまり、本明細書において説明する技法は、仮想マシン・マネージャーが、オペレーティング・システム１１２の不正利用された(compromised)コードがカーネル・モードで実行することを防止することにより、追加のレベルのセキュリティを設ける。しかしながら、同時に、本明細書において説明する技法は、ユーザー・モードで実行するコードのコード完全性を検証することを可能とし、および／またはオペレーティング・システム１１２によって実施される他のポリシーを適宜可能にする。オペレーティング・システム１１２は、ユーザー・モードで実行するコードに対してポリシーを設定し、オペレーティング・システム１１２が、コード検証が実行されない状況をサポートすることを可能にする。例えば、オペレーティング・システム１１２は、あるコードが検証されることなく

ユーザー・モードで実行させること、またはユーザー・モードで実行するあるコードのために動的コード生成を実行させることを望む場合がある。このような判定は、オペレーティング・システム１１２によって行うことができるのは、全て、仮想マシン・マネージャー１０２（または他の上位特権エンティティ）によってオペレーティング・システム１１２のコード完全性が確認されたので、オペレーティング・システム１１２は不正利用されていないことを、計算デバイス１００のユーザーが保証されている間である。

【００３１】

[0036] 図３は、１つ以上の実施形態にしたがって、仮想マシン・マネージャーによって支援される選択的コード完全性強制を実施するプロセス例３００を示すフローチャートである。プロセス３００は、図１および図２の仮想マシン・マネージャー１０２のような、仮想マシン・マネージャーによって少なくとも部分的に実行され、ソフトウェア、ファームウェア、ハードウェア、またはその組み合わせで実装することができる。プロセス３００は、１組のアクトとして示され、種々のアクトの動作を実行するために示される順序には限定されない。プロセス３００は、仮想マシン・マネージャーによって支援される選択的コード完全性強制を実施するプロセス例であり、仮想マシン・マネージャーによって支援される選択的コード完全性強制を実施することについての追加の説明が、異なる図を参照して本明細書に含まれる。

【００３２】

[0037] プロセス３００において、仮想プロセッサーによって実行される実行可能コードを含むメモリー・ページを識別する（アクト３０２）。メモリー・ページは、異なる時点において、そして、仮想マシンまたは仮想マシンによって管理されるオペレーティング・システムによるメモリー・ページを実行可能にする要求、仮想マシンまたは仮想マシンによって管理されるオペレーティング・システムによる、プログラムを仮想プロセッサーによって実行させる要求等のような、異なるイベントに応答して識別することができる。

【００３３】

[0038] メモリー・ページがカーネル・モードで実行可能にすべきか否かについて判定を行う（アクト３０４）。この判定は、メモリー・ページを実行可能にする要求を行うときに、仮想マシンまたはオペレーティング・システムによって識別されるというように、種々のやり方で行うことができる。

【００３４】

[0039] メモリー・ページがカーネル・モードで実行可能にすべきであると判定したことに応答して、実行可能コードのコード完全性チェックを、上位特権エンティティによって実行する（アクト３０６）。上位特権エンティティとは、先に説明したように、仮想マシンによって管理されるオペレーティング・システムよりも多くの特権を与えられたエンティティである。このより多くの特権を与えられたエンティティは、先に説明したように、仮想マシン・マネージャー、または他のエンティティとすることができる。コード完全性チェックは、先に説明したように種々のやり方でコード完全性ポリシーに基づいてコードの完全性を検証することによって、例えば、信頼チェーンが検証されること、およびコードが変更されていないことをディジタル署名が検証することに基づいて、コードを検証することによって行われる。コード完全性チェックが実行可能コードを確認した場合にのみ、メモリー・ページにおける実行可能コードの実行は許される（アクト３０８）。メモリー・ページにおける実行可能コードの実行は、例えば、仮想マシン・マネージャーが、アクト３０２において識別されたメモリー・ページに実行許可を与えることによって、許すことができる。

【００３５】

[0040] アクト３０４に戻り、メモリー・ページはカーネル・モードで実行可能にすべきでないと判定したことに応答して、仮想マシン・マネージャーによって管理される仮想マシンのオペレーティング・システムは、オペレーティング・システムのポリシーに基づいて、実行可能コードの実行を許すか否か判定することを許される（アクト３１０）。オペレーティング・システムは、先に説明したように、コード完全性チェックを含む、コー

ドを実行すべきか否か判定する種々の異なるポリシーを実施することができる。
【００３６】
[0041] アクト３０２における、メモリー・ページを実行可能にする要求は、仮想マシン・マネージャーに、または代わりに上位特権エンティティ（例えば、図２の上位特権エンティティ２０４）に向けて送ることができる。１つ以上の実施形態では、仮想マシン・マネージャーが上位特権エンティティである場合、要求は仮想マシン・マネージャーに向けて送られ、仮想マシン・マネージャーがアクト３０６においてコード完全性チェックを実行する。しかしながら、他のエンティティが上位特権エンティティである場合、要求はその上位特権エンティティに（例えば、直接または仮想マシン・マネージャーを介して）向けて送られ、この上位特権エンティティがアクト３０６においてコード完全性チェックを実行し、アクト３０６におけるコード完全性チェックが実行可能コードを確認した場合、上位特権エンティティは、仮想マシン・マネージャーに、メモリー・ページをカーネル・モードで実行可能にすることを通知する。仮想マシン・マネージャーは、例えば、以下で更に詳しく説明するように、第２レベル・アドレス変換表を更新することによって、メモリー・ページをカーネル・モードで実行可能にすることができる。
【００３７】
[0042] このように、仮想マシン・マネージャーは、選択的コード完全性強制を支援する。カーネル・モード・コードのコード完全性は、仮想マシン・マネージャーによって強制され（または仮想マシン・マネージャーを利用する他の上位特権エンティティ）、一方ユーザー・モード・コードのコード完全性の強制は、仮想マシン・マネージャーによって管理される仮想マシンにおいて実行するオペレーティング・システムに任せられる。
【００３８】
[0043] 図１に戻り、１つ以上の実施形態では、計算デバイス１００は仮想メモリーを採用する。仮想メモリーとは、他のアドレス空間（例えば、物理メモリー）にマッピングされるアドレス空間を指す。アプリケーションには仮想メモリー空間が割り当てられ、この仮想アドレス空間において、アプリケーション・コードが実行され、データーが格納される。メモリー・マネージャー（例えば、プロセッサーの）は、仮想メモリー空間における仮想メモリー・アドレスの、他のメモリー空間におけるアドレスへのマッピングを管理する。仮想メモリー・アドレス空間から他のメモリー空間に仮想メモリー・アドレスをマッピングするとき、アドレス変換が実行される。アドレス変換表が、このマッピングを実行するために使用され、本明細書において説明する技法を実現するために利用することができる。
【００３９】
[0044] １つ以上の実施形態では、アドレス変換表は、計算デバイス１００のコンポーネント１０４である物理プロセッサーによってというように、ハードウェアで実装される。アドレス変換表は、仮想マシン・マネージャー１０２が、以下で更に詳しく説明するように、選択的にコード完全性を強制することを可能にする。あるいは、計算デバイス１００のハードウェア（例えば、コンポーネント１０４である物理プロセッサー）が種々の他の表、リスト、レコード、構造等を使用して、仮想マシン・マネージャー１０２に選択的にコード完全性を強制させることもできる。あるいは、ソフトウェアで実装される種々の他の表、リスト、レコード、構造等（例えば、仮想マシン・マネージャー１０２の一部として、または計算デバイス１００の他のコンポーネントまたはモジュールの一部として）を使用して、仮想マシン・マネージャー１０２に選択的にコード完全性を強制させることもできる。
【００４０】
[0045] 図４は、１つ以上の実施形態にしたがって、仮想マシン・マネージャーによって支援される選択的コード完全性強制を実施するシステム例４００を示す。システム４００は、例えば、図１の計算デバイス１００とすることができる。システム４００は、物理プロセッサー４０２、物理メモリー空間４０４、仮想プロセッサー４０６、およびプログラム４０８を含む。物理プロセッサー４０２は、図１のコンポーネント１０４とすること

ができ、物理メモリー空間４０４は、図１のコンポーネント１０４とすることができ、仮想プロセッサー４０６は図１の仮想コンポーネント１１６とすることができ、プログラム４０８は図１のアプリケーション１１４またはオペレーティング・システム１１２の一部とすることができる。物理プロセッサー４０２は、物理メモリー空間４０４へのアクセスを管理するメモリー・マネージャー４１０を含む。物理メモリー空間４０４は、ＲＡＭ、フラッシュ・メモリー等のような、種々の揮発性および／または不揮発性メモリーとすることができる。

【００４１】

[0046] 物理プロセッサー４０２は、仮想マシン・メモリー空間４１２を仮想プロセッサー４０６に割り当て、第２レベル・アドレス変換表４１４を維持する。第２レベル・アドレス変換表４１４は、仮想マシン・メモリー空間４１２におけるアドレスを物理メモリー空間４０４におけるアドレスにマッピングする。所与の時点において、物理メモリー空間４０４のどのアドレスに、仮想マシン・メモリー空間４１２における特定のアドレスがマッピングするかは、変化することができ、メモリー・マネージャー４１０によって制御される。メモリー・マネージャー４１０は、マッピングを変化させることができ、複数の異なる仮想プロセッサーが物理メモリー空間４０４を共有することを可能にし、および／または種々の公開および／または企業固有の技法の内任意のものを使用して、仮想マシン・メモリー空間４１２が物理メモリー空間４０４よりも大きくなることを可能にする。

【００４２】

[0047] 仮想プロセッサー４０６は、仮想マシン・メモリー空間４１２へのアクセスを管理するメモリー・マネージャー４１６を含む。仮想プロセッサー４０６は、プログラム・メモリー空間４１８をプログラム４０８に割り当て、第１レベル・アドレス変換表４２０を維持する。第１レベル・アドレス変換表４２０は、プログラム・メモリー空間４１８におけるアドレスを仮想マシン・メモリー空間４１２におけるアドレスにマッピングする。任意の所与の時点において、仮想マシン・メモリー空間４１２のどのアドレスに、プログラム・メモリー空間４１８における特定のアドレスがマッピングするのかは、変化することができ、メモリー・マネージャー４１６によって制御される。メモリー・マネージャー４１６は、マッピングを変化させることができ、複数の異なるプログラムが仮想マシンのメモリー空間４１２を共有することを可能にし、および／または種々の公開および／または企業固有の技法の内任意のものを使用して、プログラム・メモリー空間４１８が仮想マシン・メモリー空間４１２よりも大きくなることを可能にする。

【００４３】

[0048] プログラム・メモリー空間４１８におけるアドレスへのアクセスに応答して、メモリー・マネージャー４１６は第１レベル・アドレス変換表４２０を使用して、プログラム・メモリー空間４１８におけるメモリー・アドレスを、仮想マシン・メモリー空間４１２におけるアドレスに変換する。オペレーティング・システムによって実行されるプログラム４０８のコードのアドレス、読み取ろうとするデーターがプログラム４０８によって格納されるアドレス、プログラム４０８によって書き込まれようとするデーターが格納されるアドレス等のように、アクセスは種々の異なる形態を取ることができ、

[0049] 同様に、仮想マシン・メモリー空間４１２におけるアドレスへのアクセスに応答して、メモリー・マネージャー４１０は、第２レベル・アドレス変換表４１４を使用して、仮想マシン・メモリー空間４１２におけるメモリー・アドレスを物理メモリー空間４０４におけるアドレスに変換する。アクセスは、仮想プロセッサー４０６によって実行が管理されるオペレーティング・システム・コア（例えば、カーネル）のコードのアドレス、オペレーティング・システムによって実行されるプログラム４０８のアドレス、読み取られるデーターがプログラム４０８またはオペレーティング・システム・コアによって格納されるアドレス、プログラム４０８またはオペレーティング・システム・コアによって書き込まれるデーターが格納されるべきアドレス等のように、種々の異なる形態を取ることができる。

【００４４】

[0050] １つ以上の実施形態では、第１レベル・アドレス変換表４２０および第２レベル・アドレス変換表４１４は、個々のアドレスではなく、アドレスのページをマッピングする。例えば、第１レベル・アドレス変換表４２０は、プログラム・メモリー空間４１８のページを仮想マシン・メモリー空間４１２のページにマッピングし、第２レベル・アドレス変換表４１４は、仮想マシン・メモリー空間４１２のページを物理メモリー空間４０４のページにマッピングする。あるいは、表４１４および／または表４２０は、個々に、または他のアドレスのサブページ集合体、複数のページの集合体等のように、他の分類(grouping)に基づいてアドレスをマッピングすることもできる。

【００４５】

[0051] 仮想プロセッサー４０６は、カーネル・モードまたはユーザー・モードで実行することができる。メモリー・マネージャー４１６は、カーネル・モードで実行することができるメモリー・ページのレコードを維持することができる。このレコードは、第１レベル・アドレス変換表４２０の一部として含めることができ、または代わりに他のやり方で維持することもできる。メモリー・マネージャー４１０は、メモリー・ページのレコードを、カーネル・モードで実行することができる仮想マシン・メモリー空間４１２に維持する。このレコードは、第２レベル・アドレス変換表４１４の一部として含めることができ、また代わりに他のやり方で維持することもできる。

【００４６】

[0052] 仮想プロセッサー４０６においてコードを実行するためにアクセスが行われると、実行されるコードのアドレスが第１レベル・アドレス変換表４２０によって変換されて、仮想マシン・メモリー空間４１２における変換アドレスが得られ、次いでこの変換アドレスが第２レベル・アドレス変換表４１４によって変換されて、物理メモリー空間４０４におけるアドレスが得られる。仮想プロセッサー４０６がユーザー・モードで動作している間にアクセスが行われた場合、仮想プロセッサー４０６において実行しているオペレーティング・システムは、コードを実行できるか否か判定するときにしかるべきポリシーを適用する。

【００４７】

[0053] しかしながら、仮想プロセッサーがカーネル・モードで動作しているときにアクセスが行われた場合、カーネル・モードに対する実行許可がページに与えられているか否かについてチェックを行う。カーネル・モードに対する実行許可が既にページに与えられているか否かの記録は、任意に、第２レベル・アドレス変換表４１４に維持することができ、または代わりに物理プロセッサー４０２によってどこかに維持することができる。このような状況では、カーネル・モードに対する実行許可が既にページに与えられている場合、カーネル・モードに対する実行許可がページに与えられているという指示を仮想プロセッサー４０６に返すことができ、仮想プロセッサー４０６において実行しているオペレーティング・システムは、仮想プロセッサー４０６がカーネル・モードにある間に、コードが実行することを許す。

【００４８】

[0054] カーネル・モードに対する実行許可が未だページに与えられていない場合、仮想プロセッサー４０６において実行しているオペレーティング・システムは、任意に、ページをカーネル・モードで実行可能にすることを要求することができる。仮想プロセッサー４０６は、先に説明したように、コード完全性チェックを実行する仮想マシン・マネージャー（または他の上位特権エンティティ）に要求を渡す。コード完全性チェックが成功した場合、ページにはカーネル・モードに対する実行許可が与えられ、コード完全性チェックが失敗した場合、ページにはカーネル・モードに対する実行許可が与えられない。

【００４９】

[0055] １つ以上の実施形態では、カーネル・モードに対する実行許可を、ユーザー・モードに対する実行許可またはポリシーとは別に指定することを可能にするために、第２レベル・アドレス変換表４１４に属性が含まれる。この属性は、種々の異なるやり方で第２レベル・アドレス変換表４１４にエンコードすることができるが、以下の３つの特性を

含む。第１に、第２レベル変換毎に独立して属性を制御できるように、第２レベル・アドレス変換表に属性を組み入れる（第２レベル・アドレス変換表４１４を使用してメモリー・マネージャー４１０によって実行される各変換）。第２に、属性はカーネル・モード実行許可に適用される。属性は、カーネル・モードにおける（そしてユーザー・モードにおける）リードおよび／またはライト許可に影響を及ぼす必要はない（が、あるいは影響を及ぼすこともできる）。第３に、属性は、ユーザー・モード実行から独立して、カーネル・モード実行を使用不可にさせて、少なくとも以下の２つの実行の組み合わせを許す。１）カーネル・モードでの実行を許さないが、ユーザー・モードでの実行を許す。２）カーネル・モードでの実行を許さず、ユーザー・モードでの実行も許さない。

【００５０】

[0056] 少なくとも１つの属性を第２レベル・アドレス変換表４１４に種々の異なるやり方で含めることができる。例えば、第２レベル・アドレス変換表における各エントリが２ビットを含みことができ、１ビットがユーザー・モード実行に対応し、１ビットがカーネル・モード実行に対応する。ユーザー・モード・ページに対応するビットには、ユーザー・モードでページにおいてコードを実行する許可が与えられたことを示すために１つの値（例えば、「１」の値を割り当てる。ビットがセットされたとも言う）を割り当てることができ、ユーザー・モードでページにおいてコードを実行する許可が与えられないことを示すために他の値（例えば、「０」の値を割り当てる。ビットがクリアされたとも言う）を割り当てることができる。同様に、カーネル・モード実行に対応するビットには、カーネル・モードでページにおいてコードを実行する許可が与えられたことを示すために、１つの値を割り当てることができ（例えば、「１」の値を割り当てる。ビットがセットされたとも言う）、カーネル・モードでページにおいてコードを実行する許可が与えられないことを示すために、他の値を割り当てることができる（例、「０」の値を割り当てる。クリアされたビットとも呼ぶ）。

【００５１】

[0057] １つ以上の実施形態では、第２レベル・アドレス変換表４１４において、メモリー・ページに対する初期値またはデフォルト値は、カーネル・モードに対する実行を許さない（実行許可が与えられない）が、ユーザー・モードに対する実行を許す（実行許可が与えられる）。メモリー・ページにおけるコードのコード完全性が仮想マシン・マネージャー（または他のより多くの特権が与えられたエンティティ）によって確認されている場合、そのページに対してはユーザー・モードだけでなくカーネル・モードの実行も許す（実行許可が与えられる）ように、メモリー・ページに対する値を変更することができる。しかしながら、メモリー・ページにおけるコードのコード完全性がオペレーティング・システムによるコード完全性チェックに失敗した場合、オペレーティング・システムは、任意に、そのページに対してユーザー・モードに対する実行を許さない（実行許可が与えられない）ように、メモリー・ページに対する値を変更する（または仮想マシン・マネージャーが変更することを要求する）ことができる。

【００５２】

[0058] 尚、第２レベル・アドレス変換表４１４は、図１の仮想マシン・マネージャー１０２のような、仮想マシン・マネージャーの制御下にあることは注記してしかるべきである。第２レベル・アドレス変換表４１４は、仮想マシンにおいて実行するソフトウェア（例えば、図１の仮想マシン１０６のアプリケーション１１４またはオペレーティング・システム１１２）によってアクセス可能ではない。仮想マシンにおけるオペレーティング・システムは、その第１レベル・アドレス変換表に対して直接制御を行い、仮想マシン・マネージャーは、第２レベル・アドレス変換表に対して排他的制御を行う。つまり、第２レベル・アドレス変換表においてカーネル・モード実行の制御を行うことにより、仮想マシン・マネージャーは効率的にそして容易にこの制御を使用して、第１レベル・アドレス変換表のオペレーティング・システムの管理とは独立して、カーネル・モード実行アクセスを付与するまたは拒否することができる。第２レベル・アドレス変換表においてカーネル・モード・コードの実行を制限する能力を有することにより、仮想マシン・マネージャ

ーは、仮想マシンにおけるオペレーティング・システムによって実行される第１レベル・アドレス変換表の更新にそれ自体を直接巻き込むことなく、そのコード完全性ポリシーを強制することができる。

【００５３】

[0059] また、以上では第１および第２レベル・アドレス変換表について説明したが、代わりに、仮想マシン・マネージャー強制コード完全性は、１つのアドレス変換表（例えば、第１レベル・アドレス変換表）を使用してソフトウェアで実現できることも注記してしかるべきである。例えば、仮想マシン・マネージャーが第１レベル・アドレス変換表（例えば、図４の表４２０）の制御を行い、仮想マシン内部からのオペレーティング・システムが直接第１レベル・アドレス変換表にアクセスすることを禁止することができる。次いで、仮想マシン・マネージャーは、先に説明したポリシーを強制することができる（例えば、仮想マシン・マネージャーは、カーネルによって実行可能なコードは、仮想マシン・マネージャーが確認したコードであることを保証する(ensure)ことができるが、仮想マシン・マネージャーはユーザー・モード・コード実行に対して何ら制限を強制しない）。

【００５４】

[0060] 本明細書では、特定のモジュールを参照して特定の機能について説明したが、本明細書において説明した個々のモジュールの機能は、複数のモジュールに分けることができ、および／または複数のモジュールの少なくとも一部の機能は、１つのモジュールに組み合わせることができることは注記してしかるべきである。加えて、本明細書においてアクションを実行すると説明した特定のモジュールは、そのアクションを実行するその特定のモジュール自体を含み、あるいは代わりに他のコンポーネントを呼び出すまたそうでなければアクセスするその特定のモジュール、またはそのアクションを実行する（またはその特定のモジュールに関連付けてアクションを実行する）モジュールを含む。つまり、アクションを実行する特定のモジュールは、そのアクションを実行するその特定のモジュール自体、および／またはそのアクションを実行するその特定のモジュールによって呼び出されるまたそうでなければアクセスされる他のモジュールも含む。

【００５５】

[0061] 図５は、本明細書において説明した種々の技法を実現することができる１つ以上のシステムおよび／または計算デバイス５０２を代表する計算デバイス例５０２を含むシステム例を、全体的に５００で示す。計算デバイス５０２は、例えば、サービス・プロバイダーのサーバー、クライアントに関連するデバイス（例えば、クライアント・デバイス）、オンチップ・システム、および／または任意の他の適した計算デバイスまたは計算システムであってもよい。

【００５６】

[0062] 図示する計算デバイス例５０２は、処理システム５０４、１つ以上のコンピューター読み取り可能媒体５０６、および１つ以上のＩ／Ｏインターフェース５０８を含み、互いに通信可能に結合されている。図示しないが、計算デバイス５０２は、更に、種々のコンポーネントを互いに結合するシステム・バスまたは他のデーターおよびコマンド転送システムも含むことができる。システム・バスは、メモリー・バスまたはメモリー・コントローラー、周辺バス、ユニバーサル・シリアル・バス、および／または種々のバス・アーキテクチャーの内任意のものを利用するプロセッサーまたはローカル・バスというような、異なるバス構造の内任意の１つまたはその組み合わせを含むことができる。制御およびデーター・ラインというような、種々の他の例も考えられる。

【００５７】

[0063] 処理システム５０４は、ハードウェアを使用して１つ以上の動作を実行する機能を表す。したがって、処理システム５０４は、プロセッサー、機能ブロック等として構成することができるハードウェア・エレメント５１０を含むことが図示されている。これは、特定用途集積回路、または１つ以上の半導体を使用して形成される他のロジック・デバイスのような、ハードウェアの実現例を含むことができる。ハードウェア・エレメント５１０は、これらが形成される材料にも、その内部で採用される処理メカニズムにも限定

されない。例えば、プロセッサーは、半導体（１つまたは複数）および／またはトランジスター（例えば、電子集積回路（ＩＣ））で構成されてもよい。このようなコンテキストでは、プロセッサー実行可能命令は電子的実行可能命令でもよい。

【００５８】

[0064] コンピューター読み取り可能媒体５０６は、メモリー／ストレージ５１２を含むことが図示されている。メモリー／ストレージ５１２は、１つ以上のコンピューター読み取り可能媒体に関連するメモリー／ストレージ容量を表す。メモリー／ストレージ５１２は、揮発性媒体（ランダム・アクセス・メモリー（ＲＡＭ）のような）、および／または不揮発性媒体（リード・オンリー・メモリー（ＲＯＭ）、フラッシュ・メモリー、光ディスク、磁気ディスク等のような）を含むことができる。メモリー／ストレージ５１２は、固定媒体（例えば、ＲＡＭ、ＲＯＭ、固定ハード・ドライブ等）、およびリムーバブル媒体（例えば、フラッシュ・メモリー、リムーバブル・ハード・ドライブ、光ディスク等）を含むことができる。コンピューター読み取り可能媒体５０６は、以下で更に説明するように、種々の他の方法で構成することができる。

【００５９】

[0065] 入力／出力インターフェース（１つまたは複数）５０８は、ユーザーにコマンドおよび情報を計算デバイス５０２に入力させる機能、および種々の入力／出力デバイスを使用してユーザーおよび／または他のコンポーネントまたはデバイスに情報を提示させる機能を代表する。入力デバイスの例には、キーボード、カーソル制御デバイス（例えば、マウス）、マイクロフォン（例えば、音声入力用）、スキャナー、タッチ機能（例えば、物理的タッチを検出するように構成された容量性センサーまたは他のセンサー）、カメラ（例えば、ジェスチャーのようにタッチを伴わない動きを検出するために、赤外線周波数のような、可視または不可視波長を採用するのでもよい）等が含まれる。出力デバイスの例には、ディスプレイ・デバイス（例えば、モニターまたはプロジェクター）、スピーカー、プリンター、ネットワーク・カード、触覚応答デバイス等が含まれる。このように、計算デバイス５０２は、ユーザーの対話処理をサポートするために、以下で更に説明するように、種々の方法で構成することができる。

【００６０】

[0066] また、計算デバイス５０２は、仮想マシン・マネージャー５１４（ハイパーバイザーとも呼ぶ）も含む。仮想マシン・マネージャー５１４は、仮想マシンが計算デバイス５０２において実行することを可能にする。仮想マシン・マネージャー５１４は、例えば、図１または図２の仮想マシン・マネージャー１０２とすることができる。

【００６１】

[0067] 本明細書では、種々の技法が、ソフトウェア、ハードウェア・エレメント、またはプログラム・モジュールという一般的なコンテキストで説明される場合がある。一般に、このようなモジュールは、ルーチン、プログラム、オブジェクト、エレメント、コンポーネント、データー構造等を含み、特定のタスクを実行するかまたは特定の抽象中層データー型を実装する。「モジュール」、「機能」、および「コンポーネント」という用語が本明細書において使用される場合、一般に、ソフトウェア、ファームウェア、ハードウェア、またはその組み合わせを表す。本明細書において説明した技法の特徴は、プラットフォーム独立ということであり、これらの技法は種々のプロセッサーを有する種々の計算プラットフォームにおいて実現できることを意味する。

【００６２】

[0068] 説明したモジュールおよび技法の実現例が、ある形態のコンピューター読み取り可能媒体に格納されること、または送信されることも可能である。コンピューター読み取り可能媒体は、計算デバイス５０２によってアクセスすることができる種々の媒体を含むことができる。一例として、そして限定ではなく、コンピューター読み取り可能媒体は「コンピューター読み取り可能記憶媒体」および「コンピューター読み取り可能信号媒体」を含むことができる。

【００６３】

[0069] 「コンピューター読み取り可能記憶媒体」とは、情報の永続的な格納、および／または単なる信号送信、搬送波、または信号自体とは対象的に、有形である格納を可能にする媒体および／またはデバイスを指す。つまり、コンピューター読み取り可能記憶媒体は、非信号支持媒体(non-signal bearing media)を指す。コンピューター読み取り可能記憶媒体は、コンピューター読み取り可能命令、データー構造、プログラム・モジュール、論理エレメント／回路、または他のデーターというような情報の格納に適した方法または技術で実現された揮発性および不揮発性、リムーバブルおよび非リムーバブル媒体、および／または記憶デバイスというようなハードウェアを含む。コンピューター読み取り可能記憶媒体の例には、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、フラッシュ・メモリーまたは他のメモリー技術、ＣＤ－ＲＯＭ、ディジタル・バーサタイル・ディスク（ＤＶＤ）または他の光ストレージ、ハード・ディスク、磁気カセット、磁気テープ、磁気ディスク記憶デバイスまたは他の磁気記憶デバイス、あるいは所望の情報を格納するのに適しておりコンピューターによってアクセスすることができる他の記憶デバイス、有形媒体、または製品を含むことができるが、これらに限定されるのではない。

【００６４】

[0070] 「コンピューター読み取り可能信号媒体」とは、ネットワークを介してというように、命令を計算デバイス５０２のハードウェアに送信するように構成された信号支持媒体を指す。信号媒体は、通例、搬送波のような変調データー信号、データー信号、または他の移送メカニズムに、コンピューター読み取り可能命令、データー構造、プログラム・モジュール、または他のデーターを具体化することができる。また、信号媒体は任意の情報配信媒体も含む。「変調データー信号」という用語は、その特性の１つ以上が、当該信号内に情報をエンコードするようなやり方で設定されたまたは変化させられた信号を意味する。一例として、そして限定ではなく、通信媒体は、有線ネットワークまたは直接有線接続のような有線媒体、および音響、ＲＦ、赤外線、および他のワイヤレス媒体のようなワイヤレス・媒体を含む。

【００６５】

[0071] 既に説明したように、ハードウェア・エレメント５１０およびコンピューター読み取り可能媒体５０６は、ハードウェア形態で実現される命令、モジュール、プログラマブル・デバイス・ロジック、および／または固定デバイス・ロジックを代表し、本明細書において説明した技法の少なくともある形態を実現するために、ある実施形態において採用することができる。ハードウェア・エレメントは、集積回路またはオンチップ・システムのコンポーネント、特定用途集積回路（ＡＳＩＣ）、フィールド・プログラマブル・ゲート・アレイ（ＦＰＧＡ）、複合プログラマブル・ロジック・デバイス（ＣＰＬＤ）、およびシリコンまたは他のハードウェア・デバイスにおける他の実現例を含むことができる。このコンテキストでは、ハードウェア・エレメントは、実行のために命令を格納するために利用されるハードウェア・エレメントおよびハードウェア・デバイス、例えば、既に説明したコンピューター読み取り可能記憶媒体によって具体化される命令、モジュール、および／またはロジックによって定められるプログラム・タスクを実行する処理デバイスとして動作することができる。

【００６６】

[0072] 以上の組み合わせも、本明細書において説明した種々の技法およびモジュールを実現するために採用されてもよい。したがって、ソフトウェア、ハードウェア、またはプログラム・モジュールおよび他のプログラム・モジュールは、ある形態のコンピューター読み取り可能記憶媒体においておよび／または１つ以上のハードウェア・エレメント５１０によって具体化される１つ以上の命令および／またはロジックとして実現されてもよい。計算デバイス５０２は、ソフトウェアおよび／またはハードウェア・モジュールに対応する特定の命令および／または機能を実現するように構成されてもよい。したがって、計算デバイス５０２によってソフトウェアとして実行可能であるモジュールとしてのモジュールの実現例は、少なくとも部分的にハードウェアによって、例えば、コンピューター読み取り可能記憶媒体および／または処理システムのハードウェア・エレメント５１０の

使用によって達成することができる。命令および／または機能は、本明細書において説明した技法、モジュール、および例を実現するために、１つ以上の製品（例えば、１つ以上の計算デバイス５０２および／または処理システム５０４）によって実行可能／動作可能であってもよい。

【００６７】

[0073] 更に図５に示すように、システム例５００は、パーソナル・コンピューター（ＰＣ）、テレビジョン・デバイス、および／または移動体デバイスにおいてアプリケーションを実行するとき、シームレスなユーザー体験のために遍在環境を可能にする。サービスおよびアプリケーションは、３つの環境全てにおいて、１つのデバイスから次に移るときに、アプリケーションを利用する、ビデオ・ゲームをプレーする、ビデオを見る等の間、共通ユーザー体験のために実質的に同様に実行する。

【００６８】

[0074] システム例５００では、多数のデバイスが中央計算デバイスを介して相互接続される。中央計算デバイスは、複数のデバイスに対してローカルであってもよく、または複数のデバイスから離れて配置されてもよい。１つ以上の実施形態では、中央計算デバイスは、ネットワーク、インターネット、または他のデーター通信リンクを介して複数のデバイスに接続された１つ以上のサーバー・コンピューターのクラウドであってもよい。

【００６９】

[0075] １つ以上の実施形態では、この相互接続アーキテクチャーは、機能を複数のデバイスに跨がって配信し、複数のデバイスのユーザーに共通で継ぎ目のない体験を提供することを可能にする。複数のデバイスの各々は、異なる物理的要件および能力を有してもよく、中央計算デバイスは、デバイスへの体験の配信を可能にするプラットフォームを使用する。このプラットフォームは、そのデバイスに合わせて作られ、しかも全てのデバイスに共通である。１つ以上の実施形態では、１つのクラスのターゲット・デバイスが作られ、この包括的なクラスのデバイスに合わせて体験が作られる。１つのクラスのデバイスは、デバイスの物理的特徴、使用の形式、または他の共通特性によって定められてもよい。

【００７０】

[0076] 種々の実現例では、計算デバイス５０２は、コンピューター５１６、移動体５１８、およびテレビジョン５２０の使用のためというような、種々の異なる構成を取ることができる。これらの構成の各々は、全体的に異なる構造および能力を有するデバイスを含み、つまり計算デバイス５０２は、これらの異なるデバイス・クラスの内１つ以上にしたがって構成されてもよい。例えば、計算デバイス５０２は、パーソナル・コンピューター、デスクトップ・コンピューター、マルチスクリーン・コンピューター、ラップトップ・コンピューター、ネットブック等を含む、コンピューター５１６クラスのデバイスとして実現することができる。

【００７１】

[0077] また、計算デバイス５０２は、移動体電話機、携帯用音楽プレーヤー、携帯用ゲーミング・デバイス、タブレット・コンピューター、マルチスクリーン・コンピューター等のような、移動体デバイスを含む移動体５１８クラスのデバイスとして実現することもできる。また、計算デバイス５０２は、日常の視聴環境における、一般にもっと大きな画面を有するデバイスまたはこれに接続されたデバイスを含むテレビジョン５２０クラスのデバイスとして実現することもできる。これらのデバイスは、テレビジョン、セット・トップ・ボックス、ゲーミング・コンソール等を含む。

【００７２】

[0078] 本明細書において説明した技法は、これら種々の構成の計算デバイス５０２によってサポートすることができ、本明細書において説明した技法の具体的な例には限定されない。また、この機能は、以下で説明するように、「クラウド」５２２上でプラットフォーム５２４を介してというように、全体的にまたは部分的に分散型システムの使用によって実現することもできる。

【００７３】

[0079] クラウド５２２は、リソース５２６のためのプラットフォーム５２４を含む、および／またはプラットフォーム５２４を代表する。プラットフォーム５２４は、クラウド５２２のハードウェア（例えば、サーバー）およびソフトウェア・リソースの基礎的機能を抽象化する。リソース５２６は、計算デバイス５０２から離れたサーバーにおいてコンピューター処理が実行されている間に利用することができるアプリケーションおよび／またはデーターを含むことができる。また、リソース５２６は、インターネット上で、および／またはセルラーまたはＷｉ－Ｆｉネットワークのような加入者ネットワークを介して提供されるサービスを含むこともできる。

【００７４】

[0080] プラットフォーム５２４は、計算デバイス５０２を他の計算デバイスと接続するためのリソースおよび機能を抽象化することができる。また、プラットフォーム５２４は、プラットフォーム５２４を介して実装されるリソース５２６に対して出された(encounter)要求量に対応する規模レベルを提供するために、リソースのスケーリングを抽象化する機能も果たすことができる。したがって、相互接続デバイスの実施形態では、本明細書において説明した機能の実現例は、システム５００にわたって分散させることができる。例えば、機能は、部分的に計算デバイス５０２において実現され、クラウド５２２の機能を抽象化するプラットフォーム５２４を介して実現されてもよい。

【００７５】

[0081] 以上、主題について構造的特徴および／または方法論的アクトに特定の文言で説明したが、添付した特許請求の範囲において定められる主題は、以上で説明した具体的な特徴やアクトに必ずしも限定されないことは理解されよう。逆に、以上で説明した具体的な特徴およびアクトは、特許請求の範囲を実現する形態例として開示したまでである。

【図１】

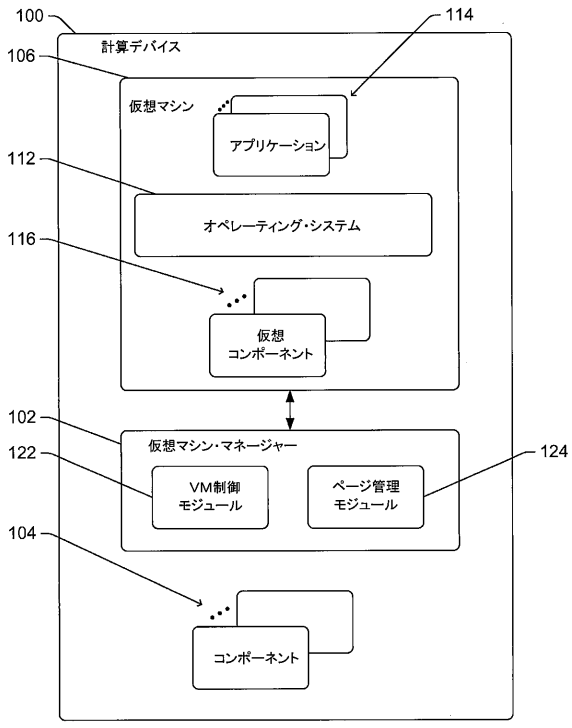

Fig. 1

【図２】

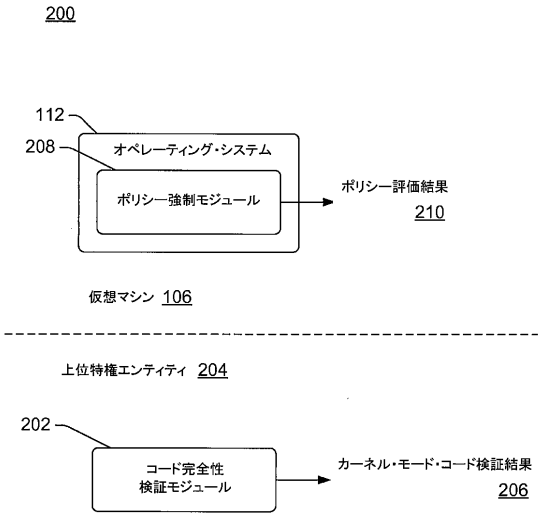

Fig. 2

【図３】

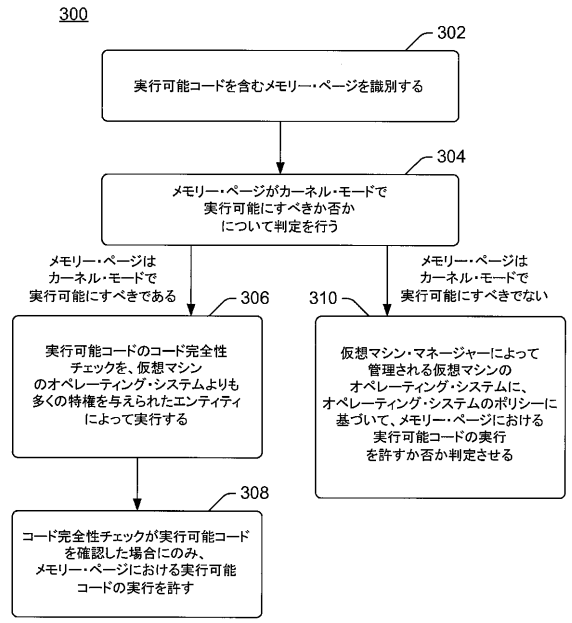

Fig. 3

【図４】

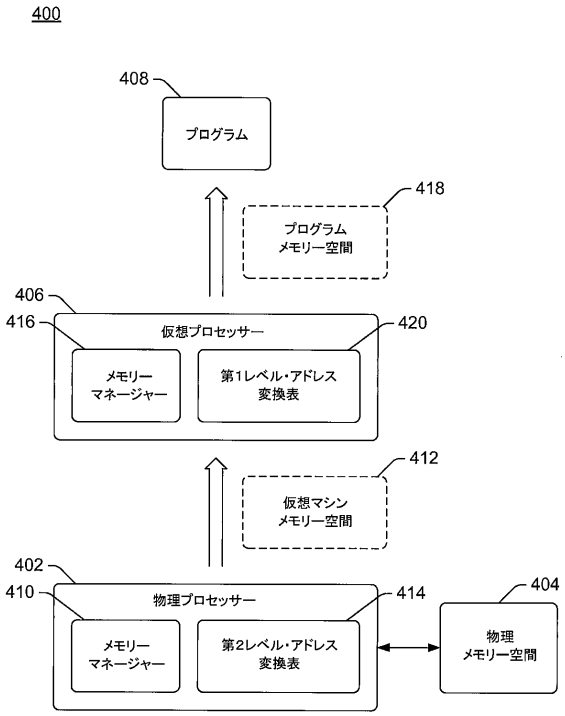

Fig. 4

【図５】

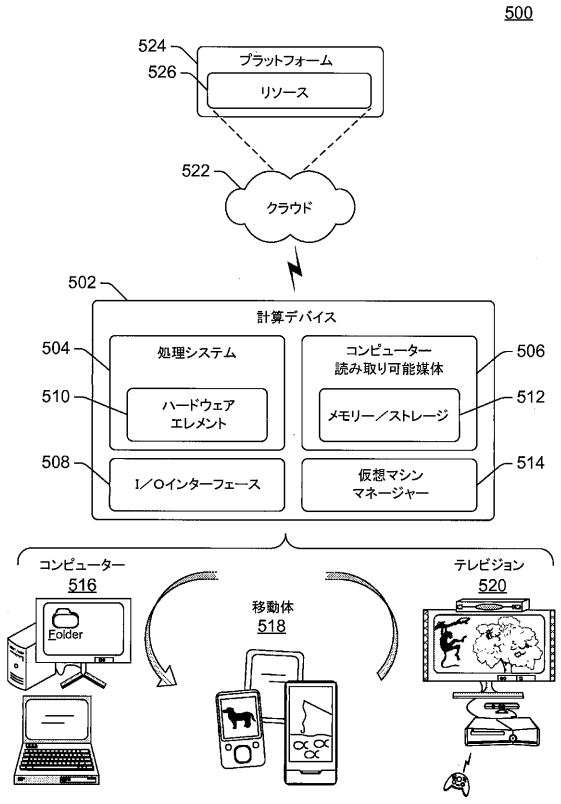

Fig. 5

フロントページの続き

(74)代理人　100120112
弁理士　中西　基晴
(72)発明者　ヘプキン，デーヴィッド・エイ
アメリカ合衆国ワシントン州９８０５２－６３９９，レッドモンド，マイクロソフト　コーポレーション，エルシーエイ－インターナショナル・パテンツ（８／１１７２）
(72)発明者　ジョンソン，ケネス・ディー
アメリカ合衆国ワシントン州９８０５２－６３９９，レッドモンド，マイクロソフト　コーポレーション，エルシーエイ－インターナショナル・パテンツ（８／１１７２）

審査官　平井　誠

(56)参考文献　特開２００９－１９９５３０（ＪＰ，Ａ）
米国特許出願公開第２０１２／０２５４９９５（ＵＳ，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／５１
Ｇ０６Ｆ　２１／６４