(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2016-511903
(P2016-511903A)
(43)公表日　平成28年4月21日(2016.4.21)

(51)Int.Cl.	F I	テーマコード（参考）
G06F 21/53 (2013.01)	G06F 21/53	

審査請求 未請求　予備審査請求 未請求　（全 24 頁）

(21)出願番号	特願2015-559213 (P2015-559213)
(86)(22)出願日	平成26年2月4日 (2014.2.4)
(85)翻訳文提出日	平成27年8月31日 (2015.8.31)
(86)国際出願番号	PCT/RO2014/000006
(87)国際公開番号	WO2014/129918
(87)国際公開日	平成26年8月28日 (2014.8.28)
(31)優先権主張番号	13/774,720
(32)優先日	平成25年2月22日 (2013.2.22)
(33)優先権主張国	米国 (US)

(71)出願人　312016539
ビットディフェンダー　アイピーアール　マネジメント　リミテッド
キプロス共和国、ニコシア、クレオントス、１２　シーワイ－１０７６
(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
(74)代理人　100196508
弁理士　松尾　淳一

最終頁に続く

(54)【発明の名称】仮想マシンの完全性保護のためのメモリイントロスペクションエンジン

(57)【要約】
開示されるシステムおよび方法はコンピュータシステムをウイルスおよびルートキットなどのマルウェアから保護することを可能にする。いくつかの実施形態において、ハイパーバイザはオペレーティングシステム（ＯＳ）の集合をホストするハードウェア仮想化プラットフォームを構成する。ハイパーバイザのプロセッサ特権レベルで実行するメモリイントロスペクションエンジンは各ＯＳを動的に識別し、保護準備モジュールを使用して、それぞれのＯＳにネイティブのメモリ割り当て関数による対象ソフトウェアオブジェクトへのメモリの割り当て方を変更する。いくつかの実施形態において、変更はマルウェア保護を必要とする対象オブジェクトのみに影響し、また対象オブジェクトのデータを含むメモリページがそれぞれのオブジェクトのために排他的に予約されるように実施することを含む。メモリイントロスペクションエンジンは次いでそれぞれのメモリページを書き込み保護する。

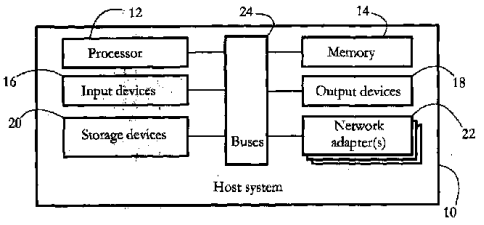

FIG. 1

【特許請求の範囲】
【請求項１】
少なくとも１つのプロセッサを備えたホストシステムであって、前記少なくとも１つのプロセッサは、
仮想マシンの仮想化物理メモリの一部分を、前記仮想マシン内で実行する対象ソフトウェアオブジェクトに割り当てるように構成されたオペレーティングシステムであって、前記仮想マシンは、ホストシステム上で実行するハイパーバイザにより公開され、前記仮想化物理メモリは複数のページに分割され、１つのページは、前記仮想化物理メモリとホストシステムの物理メモリとの間で個別にマッピングされるメモリの最小単位である、オペレーティングシステムと、
前記対象ソフトウェアオブジェクトがマルウェア保護の選択基準を満たすかどうかの判定に応答して、前記対象ソフトウェアオブジェクトが前記選択基準を満たす場合に、前記対象ソフトウェアオブジェクトのメモリ割り当てを変更するように構成された保護準備モジュールであって、前記メモリ割り当てを変更することは、前記対象ソフトウェアオブジェクトの少なくとも一部を含むどのページも前記対象ソフトウェアオブジェクトのために予約されることを保証することを含む、保護準備モジュールと
を実行するように構成された、ホストシステム。
【請求項２】
請求項１に記載のホストシステムであって、前記保護準備モジュールに接続されたメモリイントロスペクションエンジンであって、前記保護準備モジュールが前記メモリ割り当てを変更するのに応答して、前記対象ソフトウェアオブジェクトの少なくとも一部を含むすべてのページを書き込み保護するように構成されたメモリイントロスペクションエンジンをさらに備えたホストシステム。
【請求項３】
請求項２に記載のホストシステムであって、前記メモリイントロスペクションエンジンは、
前記対象ソフトウェアオブジェクトが初期化されたかどうかを判定し、
それに応答して、前記対象ソフトウェアオブジェクトが初期化された場合に、前記対象ソフトウェアオブジェクトの少なくとも一部を含むすべてのページを書き込み保護するようにさらに構成された、ホストシステム。
【請求項４】
請求項１に記載のホストシステムであって、前記選択基準は、前記対象ソフトウェアオブジェクトを前記オペレーティングシステムの種類に従って選択することを含む、ホストシステム。
【請求項５】
請求項４に記載のホストシステムであって、前記対象ソフトウェアオブジェクトがマルウェア保護の前記選択基準を満たすかどうかの前記判定は、前記仮想マシンのモデル固有レジスタ（ＭＳＲ）の内容に従って前記オペレーティングシステムの前記種類を識別することを含む、ホストシステム。
【請求項６】
請求項４に記載のホストシステムであって、前記対象ソフトウェアオブジェクトがマルウェア保護の前記選択基準を満たすかどうかの前記判定は、前記仮想マシンのモデル固有レジスタ（ＭＳＲ）により指し示されるメモリの内容に従って前記オペレーティングシステムの前記種類を識別することを含む、ホストシステム。
【請求項７】
請求項１に記載のホストシステムであって、前記メモリ割り当てを変更することは、前記オペレーティングシステムのメモリ割り当て関数をフックすることを含む、ホストシステム。
【請求項８】
請求項１に記載のホストシステムであって、前記メモリ割り当てを変更することは、前

記オペレーティングシステムのメモリ割り当て関数に、前記対象ソフトウェアオブジェクトの少なくとも一部を含むどのページも前記対象ソフトウェアオブジェクトに排他的に割り当てるように指示することを含む、ホストシステム。
【請求項９】
請求項８に記載のホストシステムであって、前記メモリ割り当て関数に指示することは、前記対象ソフトウェアオブジェクトのサイズをページサイズの整数倍に変更することを含む、ホストシステム。
【請求項１０】
請求項９に記載のホストシステムであって、前記メモリ割り当て関数に指示することは、前記一部分をページ境界に揃えることをさらに含む、ホストシステム。
【請求項１１】
請求項１に記載のホストシステムであって、前記保護準備モジュールはページの予約プールを確立するようにさらに構成され、前記プールはマルウェア保護ソフトウェアオブジェクトに割り当てるために予約され、前記メモリ割り当てを変更することはメモリページの前記予約プール内の前記一部分を割り当てることを含む、ホストシステム。
【請求項１２】
請求項１に記載のホストシステムであって、前記対象ソフトウェアオブジェクトがマルウェア保護の前記選択基準を満たすかどうかの前記判定は、
前記対象ソフトウェアオブジェクトがドライバオブジェクトであるかどうかを判定することと、
それに応答して、前記対象ソフトウェアオブジェクトがドライバオブジェクトである場合に、前記対象ソフトウェアオブジェクトがマルウェア保護の前記選択基準を満たすと判定することと
を含む、ホストシステム。
【請求項１３】
請求項１に記載のホストシステムであって、前記保護準備モジュールは前記対象ソフトウェアオブジェクトの割り当て解除を変更するようにさらに構成され、前記割り当て解除を変更することは、
前記対象ソフトウェアオブジェクトの少なくとも一部を含むページが書き込み保護されているかどうかを判定することと、
それに応答して、前記ページが書き込み保護されている場合に、前記ページの書き込み保護を解除することと
を含む、ストシステム。
【請求項１４】
請求項１２に記載のホストシステムであって、前記割り当て解除を変更することは、前記オペレーティングシステムのメモリ割り当て解除関数をフックすることをさらに含む、ホストシステム。
【請求項１５】
仮想マシンの仮想化物理メモリの一部分を、前記仮想マシン内で実行する対象ソフトウェアオブジェクトに割り当てるように構成されたオペレーティングシステムを形成するために、ホストシステムの少なくとも１つのプロセッサを使用するステップであって、前記仮想マシンは、前記ホストシステム上で実行するハイパーバイザにより公開され、前記仮想化物理メモリは複数のページに分割され、１つのページは、前記仮想化物理メモリと前記ホストシステムの物理メモリとの間で個別にマッピングされるメモリの最小単位である、ステップと、
前記対象ソフトウェアオブジェクトがマルウェア保護の選択基準を満たすかどうかの判定に応答して、前記対象ソフトウェアオブジェクトが前記選択基準を満たす場合に、前記対象ソフトウェアオブジェクトのメモリ割り当てを変更するために、前記少なくとも１つのプロセッサを使用するステップであって、前記メモリ割り当てを変更することは、前記対象ソフトウェアオブジェクトの少なくとも一部を含むどのページも前記対象ソフトウェ

アオブジェクトのために予約されることを保証することを含む、ステップと
を含む方法。
【請求項１６】
請求項１５に記載の方法であって、前記メモリ割り当てを変更するのに応答して、前記対象ソフトウェアオブジェクトの少なくとも一部を含むすべてのページを書き込み保護するステップをさらに含む方法。
【請求項１７】
請求項１６に記載の方法であって、前記対象ソフトウェアオブジェクトが初期化されたかどうかを判定するステップと、
それに応答して、前記対象ソフトウェアオブジェクトが初期化された場合に、前記対象ソフトウェアオブジェクトの少なくとも一部を含むすべてのページを書き込み保護するステップと
をさらに含む方法。
【請求項１８】
請求項１５に記載の方法であって、前記選択基準は、前記対象ソフトウェアオブジェクトを前記オペレーティングシステムの種類に従って選択するステップを含む、方法。
【請求項１９】
請求項１８に記載の方法であって、前記対象ソフトウェアオブジェクトがマルウェア保護の前記選択基準を満たすかどうかの前記判定は、前記仮想マシンのモデル固有レジスタ（ＭＳＲ）の内容に従って前記オペレーティングシステムの前記種類を識別するステップを含む、方法。
【請求項２０】
請求項１８に記載の方法であって、前記対象ソフトウェアオブジェクトがマルウェア保護の前記選択基準を満たすかどうかの前記判定は、前記仮想マシンのモデル固有レジスタ（ＭＳＲ）により指し示されるメモリの内容に従って前記オペレーティングシステムの前記種類を識別するステップを含む、方法。
【請求項２１】
請求項１５に記載の方法であって、前記メモリ割り当てを変更するステップは、前記オペレーティングシステムのメモリ割り当て関数をフックするステップを含む、方法。
【請求項２２】
請求項２１に記載の方法であって、前記メモリ割り当てを変更するステップは、前記オペレーティングシステムのメモリ割り当て関数に、前記対象ソフトウェアオブジェクトの少なくとも一部を含むすべてのページを前記対象ソフトウェアオブジェクトに排他的に割り当てるように指示するステップを含む、方法。
【請求項２３】
請求項２２に記載の方法であって、前記メモリ割り当て関数に指示するステップは、前記対象ソフトウェアオブジェクトのサイズをページサイズの整数倍に変更するステップを含む、方法。
【請求項２４】
請求項２３に記載の方法であって、前記メモリ割り当て関数に指示するステップは、前記一部分をページ境界に揃えるステップをさらに含む、方法。
【請求項２５】
請求項１５に記載の方法であって、前記メモリ割り当てを変更するステップは、
ページの予約プールを確立するステップであって、前記プールはマルウェア保護ソフトウェアオブジェクトに割り当てるために予約される、ステップと、
メモリページの前記予約プール内の前記一部分を割り当てるステップと
を含む、方法。
【請求項２６】
請求項１５に記載の方法であって、前記対象ソフトウェアオブジェクトがマルウェア保護の前記選択基準を満たすかどうかの前記判定は、

前記対象ソフトウェアオブジェクトがドライバオブジェクトであるかどうかを判定するステップと、
それに応答して、前記対象ソフトウェアオブジェクトがドライバオブジェクトである場合に、前記対象ソフトウェアオブジェクトがマルウェア保護の前記選択基準を満たすと判定するステップと
を含む、方法。
【請求項２７】
請求項１５に記載の方法であって、前記対象ソフトウェアオブジェクトの割り当て解除を変更するために、前記少なくとも１つのプロセッサを使用するステップをさらに含み、前記割り当て解除を変更するステップは、
前記対象ソフトウェアオブジェクトの少なくとも一部を含むページが書き込み保護されているかどうかを判定するステップと、
それに応答して、前記ページが書き込み保護されている場合に、前記ページの書き込み保護を解除するステップと
を含む、方法。
【請求項２８】
請求項２７に記載の方法であって、前記割り当て解除を変更するステップは、前記オペレーティングシステムのメモリ割り当て解除関数をフックするステップをさらに含む、方法。
【請求項２９】
命令を符号化している非一時的コンピュータ可読媒体であって、前記命令は、ホストシステムの少なくとも１つのプロセッサにより実行されたときに、前記少なくとも１つのプロセッサに、
仮想マシンの仮想化物理メモリの一部分を、前記仮想マシン内で実行する対象ソフトウェアオブジェクトに割り当てることであって、前記仮想マシンは、前記ホストシステム上で実行するハイパーバイザにより公開され、前記仮想化物理メモリは複数のページに分割され、１つのページは、前記仮想化物理メモリと前記ホストシステムの物理メモリとの間で個別にマッピングされるメモリの最小単位である、割り当てることと、
前記対象ソフトウェアオブジェクトがマルウェア保護の選択基準を満たすかどうかの判定に応答して、前記対象ソフトウェアオブジェクトが前記選択基準を満たす場合に、前記対象ソフトウェアオブジェクトのメモリ割り当てを変更することであって、前記メモリ割り当てを変更することは、前記対象ソフトウェアオブジェクトの少なくとも一部を含むどのページも前記対象ソフトウェアオブジェクトのために予約されることを保証することを含む、変更することと
を実行させる、非一時的コンピュータ可読媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
[0001]本発明はコンピュータシステムをマルウェアから保護するためのシステムおよび方法に関し、特にハードウェア仮想化技術を採用するマルウェア対策システムに関する。
【背景技術】
【０００２】
[0002]マルウェアとしても知られる悪意のあるソフトウェアは世界中で多数のコンピュータシステムに影響を及ぼす。コンピュータウイルス、ワーム、およびルートキットなどの多くの形態で、マルウェアは深刻なリスクを何百万人ものコンピュータユーザにもたらし、とりわけデータや機密情報の損失、個人情報盗難、および生産性の損失に対して無防備にしてしまう。
【０００３】
[0003]ハードウェア仮想化技術は、様々な方法で物理コンピュータシステムとして作動する、一般に仮想マシンとして知られる模擬コンピュータ環境の作成を可能にする。サー

バ統合やサービスとしてのインフラストラクチャ（ＩＡＡＳ）などの典型的な用途では、いくつかの仮想マシンを同じ物理マシン上で同時に実行して、それらの間でハードウェア資源を共有してよく、したがって投資および運転コストが削減される。各仮想マシンは独自のオペレーティングシステムおよび／またはソフトウェアアプリケーションを他の仮想マシンとは別々に実行してよい。マルウェアの着実な拡散により、そのような環境で動作する各仮想マシンは潜在的にマルウェア保護を必要とする。

【０００４】

[0004]当該技術分野で一般に使用される仮想化解決策は、計算用ハードウェアと仮想マシンのオペレーティングシステム（ＯＳ：Ｏｐｅｒａｔｉｎｇ　Ｓｙｓｔｅｍ）との間で動作するソフトウェア層から成り、それぞれのＯＳよりもより多くのプロセッサ特権を有する、仮想マシンモニタとしても知られるハイパーバイザを備える。マルウェア対策動作はハイパーバイザの特権レベルで行ってよい。そのような構成はマルウェア検出および防止を容易にするかもしれないが、それらは追加で一層の複雑性をもたらすし、かなりの計算コストを伴うかもしれない。

【発明の概要】

【０００５】

[0005]ハードウェア仮想化プラットフォーム用マルウェア対策解決策であって、堅牢で拡張可能な解決策を最低の計算オーバーヘッドで開発することにかなりの関心がある。

[0006]一態様によれば、ホストシステムは、オペレーティングシステムと保護準備モジュールとを実行するように構成された少なくとも１つのプロセッサを備える。オペレーティングシステムは、ホストシステム上で実行するハイパーバイザにより公開される仮想マシンの仮想化物理メモリの一部分を、仮想マシン内で実行する対象ソフトウェアオブジェクトに割り当てるように構成される。仮想化物理メモリは、各々仮想化物理メモリとホストシステムの物理メモリとの間で個別にマッピングされるメモリの最小単位であるページに分割される。保護準備モジュールは、対象ソフトウェアオブジェクトがマルウェア保護の選択基準を満たすかどうかの判定に応答して、対象ソフトウェアオブジェクトが選択基準を満たす場合に、対象オブジェクトのメモリ割り当てを変更するように構成され、メモリ割り当てを変更することは、対象ソフトウェアオブジェクトの少なくとも一部を含むどのページも対象ソフトウェアオブジェクトのために予約されることを保証することを含む。

【０００６】

[0007]別の態様によれば、方法は、ホストシステム上で実行するハイパーバイザにより公開される仮想マシンの仮想化物理メモリの一部分を、仮想マシン内で実行する対象ソフトウェアオブジェクトに割り当てるように構成されたオペレーティングシステムを形成するために、ホストシステムの少なくとも１つのプロセッサを利用することを含む。仮想化物理メモリは、各々仮想化物理メモリとホストシステムの物理メモリとの間で個別にマッピングされるメモリの最小単位であるページに分割される。方法はさらに、対象ソフトウェアオブジェクトがマルウェア保護の選択基準を満たすかどうかの判定に応答して、対象ソフトウェアオブジェクトが選択基準を満たす場合に、対象ソフトウェアオブジェクトのメモリ割り当てを変更するために、少なくとも１つのプロセッサを利用することを含み、メモリ割り当てを変更することは、対象ソフトウェアオブジェクトの少なくとも一部を含むどのページも対象ソフトウェアオブジェクトのために予約されることを保証することを含む。

【０００７】

[0008]別の態様によれば、非一時的コンピュータ可読媒体は、ホストシステムの少なくとも１つのプロセッサにより実行されたとき少なくとも１つのプロセッサに、ホストシステム上で実行するハイパーバイザにより公開される仮想マシンの仮想化物理メモリの一部分を、仮想マシン内で実行する対象ソフトウェアオブジェクトに割り当てさせる命令を符号化している。仮想化物理メモリは、各々仮想化物理メモリとホストシステムの物理メモリとの間で個別にマッピングされるメモリの最小単位であるページに分割される。命令は

さらに少なくとも１つのプロセッサに、対象ソフトウェアオブジェクトがマルウェア保護の選択基準を満たすかどうかの判定に応答して、対象ソフトウェアオブジェクトのメモリ割り当てを変更させる。メモリ割り当ての変更は、対象ソフトウェアオブジェクトが選択基準を満たす場合に行われる。メモリ割り当てを変更することは、対象ソフトウェアオブジェクトの少なくとも一部を含むどのページも対象ソフトウェアオブジェクトのために予約されることを保証することを含む。

【０００８】

[0009]本発明の前述の態様および利点は、以下の詳細な説明を読むことでかつ図面を参照することでより良く理解されるであろう。

【図面の簡単な説明】

【０００９】

【図１】[0010]本発明のいくつかの実施形態に係るマルウェアから保護されるホストコンピュータシステムの例示的なハードウェア構成を示す。

【図２】[0011]本発明のいくつかの実施形態に係る、図１のホストシステム上で実行するハイパーバイザにより公開される仮想マシンの例示的な集合を示す。

【図３】[0012]本発明のいくつかの実施形態に係る、マルウェアから保護されるオブジェクトの集合を含む、様々なプロセッサ特権レベルでホストシステム上で実行するソフトウェアオブジェクトの例示的な階層を例示する。

【図４】[0013]本発明のいくつかの実施形態に係る、図２のシステム構成でのメモリアドレスの例示的なマッピングを示す。

【図５】[0014]仮想化物理メモリ空間がページに分割され、メモリの別個の部分が複数のソフトウェアオブジェクトの各々に割り当てられる、例示的なメモリ割り当てを示す。

【図６】[0015]ページが本発明のいくつかの実施形態に係る保護ソフトウェアオブジェクトに排他的に割り当てられる、例示的なメモリ割り当てを例示する。

【図７】[0016]本発明のいくつかの実施形態に係る仮想マシンを保護するためにメモリイントロスペクションエンジンにより実施されるステップの例示的なシーケンスを示す。

【図８】[0017]対象ソフトウェアオブジェクトに対するメモリの割り当てを改変するために保護準備モジュールの実施形態により実施されるステップの例示的なシーケンスを示す。

【図９】[0018]対象ソフトウェアオブジェクトに対するメモリの割り当てを改変するために保護準備モジュールの別の実施形態により実施されるステップの例示的なシーケンスを示す。

【図１０】[0019]本発明のいくつかの実施形態に係る対象ソフトウェアオブジェクトに対するメモリの割り当て解除を改変するために保護準備モジュールにより実施されるステップの例示的なシーケンスを示す。

【発明を実施するための形態】

【００１０】

[0020]以下の説明では、すべての詳述される構造間接続は直接作用接続または中間構造を介する間接作用接続であり得ると解釈する。要素の集合は１つまたは複数の要素を含む。任意の要素の詳述は少なくとも１つの要素を指すと解釈する。複数の要素は少なくとも２つの要素を含む。別途要求しない限り、任意の説明される方法ステップは必ずしも特定の例示の順序で実施する必要はない。第１の要素（例えばデータ）は、第２の要素から導出されている場合、第２の要素に等しい第１の要素に加え、第２の要素および任意選択で他のデータを処理することにより生成される第１の要素をも包含する。パラメータに従って判定または決定を下すことは、パラメータに従っておよび任意選択で他のデータに従って判定または決定を下すことを包含する。別途特記しない限り、或る量／データの標識は量／データ自体、または量／データ自体とは異なる標識であってよい。別途特記しない限り、ページはホストシステムの物理メモリに個別にマッピングされる仮想化物理メモリの最小単位を表す。別途特記しない限り、仮想化物理メモリの一部分は、ページの集合の各ページがそれぞれの部分の一部を含む場合、ページの集合に及ぶと言う。対象オブジェク

トのためにページを予約（確保）することは、ページ全体を対象オブジェクトに割り当てること、およびそうでなければそれぞれのページが対象オブジェクトとは別個のオブジェクトの一部をホストしないことを保証することを包含する。コンピュータ可読媒体は磁気、光学、および半導体記憶媒体（例えばハードドライブ、光ディスク、フラッシュメモリ、ＤＲＡＭ）などの非一時的媒体に加え、伝導ケーブルおよび光ファイバリンクなどの通信リンクをも包含する。いくつかの実施形態によれば、本発明はとりわけ、本明細書に記載の方法を実施するようにプログラムされるハードウェア（例えば１つまたは複数のプロセッサ）を備えるコンピュータシステムに加え、本明細書に記載の方法を実施するための命令を符号化したコンピュータ可読媒体をも提供する。

【００１１】

[0021]以下の説明は必ずしも限定としてではなく例として本発明の実施形態を例示する。

[0022]図１は、本発明のいくつかの実施形態に係るホストシステム１０の例示的なハードウェア構成を示す。ホストシステム１０は企業サーバなどの法人計算用装置、またはとりわけパーソナルコンピュータもしくはスマートフォンなどのエンドユーザ装置を表してよい。他のホストシステムには、テレビやゲーム機などの娯楽装置、またはメモリと仮想化をサポートするプロセッサとを有し、かつマルウェア保護を必要とする任意の他の装置が含まれる。図１は例示目的のコンピュータシステムを示すため、移動電話またはタブレットなどの他のクライアント装置は異なる構成を有してよい。いくつかの実施形態において、システム１０は物理装置の集合を備え、すべてバス２４の集合により接続されるプロセッサ１２、メモリユニット１４、入力装置１６の集合、出力装置１８の集合、記憶装置２０の集合、およびネットワークアダプタ２２の集合を含む。

【００１２】

[0023]いくつかの実施形態において、プロセッサ１２は信号および／またはデータの集合で計算および／または論理演算を実行するように構成された物理装置（例えばマルチコア集積回路）を備える。いくつかの実施形態において、そのような論理演算は一連のプロセッサ命令（例えばマシンコードまたは他の種類のソフトウェア）の形態でプロセッサ１２に送達される。メモリユニット１４は命令を実行する過程でプロセッサ１２によりアクセスまたは生成されるデータ／信号を記憶する揮発性コンピュータ可読媒体（例えばＲＡＭ）を備えてよい。入力装置１６は、ユーザがデータおよび／または命令をシステム１０に導入できるようにするそれぞれのハードウェアインタフェースおよび／またはアダプタを含む、とりわけコンピュータキーボード、マウス、およびマイクロホンを含んでよい。出力装置１８は、とりわけモニタなどの表示装置およびスピーカに加え、システム１０がデータをユーザに伝達できるようにする、グラフィックカードなどのハードウェアインタフェース／アダプタをも含んでよい。いくつかの実施形態において、入力装置１６および出力装置１８はタッチスクリーン装置の場合のようにハードウェアの共通部分を共有してよい。記憶装置２０はソフトウェア命令および／またはデータの不揮発性記憶、読み取り、および書き込みを可能にするコンピュータ可読媒体を含む。例示的な記憶装置２０は磁気および光ディスクならびにフラッシュメモリ装置に加えて、ＣＤおよび／またはＤＶＤディスクなどの取り外し可能媒体ならびにドライブをも含む。ネットワークアダプタ２２の集合はシステム１０がコンピュータネットワークにおよび／または他の装置／コンピュータシステムに接続できるようにする。バス２４は複数のシステム、周辺、およびチップセットバス、ならびに／またはホストシステム１０の装置１２～２２の相互通信を可能にするすべての他の回路を集合的に表す。例えばバス２４は、プロセッサ１２をメモリ１４に接続するノースブリッジ、および／またはプロセッサ１２をとりわけ装置１６～２２に接続するサウスブリッジを備えてよい。

【００１３】

[0024]図２は、ホストシステム１０上で実行し、本発明のいくつかの実施形態に係るハイパーバイザ３０により公開されるゲスト仮想マシン３２ａ～ｂの例示的な集合を示す。仮想マシン（ＶＭ：Ｖｉｒｔｕａｌ　Ｍａｃｈｉｎｅ）は一般に当該技術分野で実際の物

理マシン／コンピュータシステムのソフトウェアエミュレーションとして知られ、各々独自のオペレーティングシステムとソフトウェアを他のＶＭとは独立して実行することができる。ハイパーバイザ３０は複数の仮想マシンによる、プロセッサ動作、メモリ、記憶、入出力、およびネットワーキング装置などのホストシステム１０のハードウェア資源の多重化（共有化）を可能にするソフトウェアを備える。いくつかの実施形態において、ハイパーバイザ３０は複数の仮想マシンおよび／またはオペレーティングシステム（ＯＳ）がホストシステム１０上で様々な分離度で同時に実行できるようにする。そのような構成を可能にするために、ハイパーバイザ３０のソフトウェア形成部が、各々とりわけプロセッサ１２やメモリ１４などのシステム１０の物理ハードウェア装置をエミュレートする、複数の仮想化、すなわちソフトウェアエミュレートした装置を作成してよい。ハイパーバイザ３０はさらに仮想装置の集合をホストシステム１０上で動作する各ＶＭに割り当ててよい。したがって各ＶＭ３２ａ～ｂは自身の物理装置の集合を所有するかのように、すなわちほぼ完全なコンピュータシステムとして動作する。普及しているハイパーバイザの例には、とりわけＶＭｗａｒｅ　Ｉｎｃ．のＶＭｗａｒｅ　ｖＳｐｈｅｒｅ（商標）、およびオープンソースのＸｅｎハイパーバイザが含まれる。

【００１４】

[0025]いくつかの実施形態において、ハイパーバイザ３０は、さらに後述されるようにマルウェア対策動作を実施するように構成されたメモリイントロスペクションエンジン４０、およびメモリイントロスペクションエンジン４０に接続される保護準備モジュール４６を含む。エンジン４０およびモジュール４６はハイパーバイザ３０に組み込まれてよく、またはハイパーバイザ３０とは別個で独立しているがハイパーバイザ３０と実質的に同じプロセッサ特権レベルで実行するソフトウェアコンポーネントとして設けてよい。単一のエンジン４０をホストシステム１０上で実行する複数のＶＭをマルウェア保護するように構成してよい。

【００１５】

[0026]図２は簡略化のため２つのＶＭ３２ａ～ｂのみを示すが、ホストシステム１０は多数、例えば何百ものＶＭを動作させてよく、またそのようなＶＭの数はホストシステム１０の動作中に変化してよい。いくつかの実施形態において、各ＶＭ３２ａ～ｂはそれぞれゲストオペレーティングシステム３４ａ～ｂおよび／またはソフトウェアアプリケーション４２ａ～ｂおよび４２ｃ～ｄの集合を、同時にかつホストシステム１０上で実行する他のＶＭとは独立して実行する。各ＯＳ３４ａ～ｂはそれぞれのＶＭ３２ａ～ｂの（仮想化）ハードウェアへのインタフェースを提供するソフトウェアを備え、またそれぞれ、それぞれのＯＳ上で実行するアプリケーション４２ａ～ｂおよび４２ｃ～ｄを計算するためのホストとして働く。オペレーティングシステム３４ａ～ｂはとりわけＷｉｎｄｏｗｓ（登録商標）、ＭａｃＯＳ（登録商標）、Ｌｉｎｕｘ（登録商標）、ｉＯＳ（登録商標）、またはアンドロイド（商標）などの任意の広く入手可能なオペレーティングシステムを備えてよい。アプリケーション４２ａ～ｄはとりわけ文書処理、画像処理、データベース、ブラウザ、電子通信アプリケーション、およびマルウェア対策アプリケーションを含んでよい。

【００１６】

[0027]図３は、本発明のいくつかの実施形態に係るホストシステム１０上で実行するソフトウェアオブジェクトの階層を例示する。図３は当該技術分野で層または保護リングとしても知られるプロセッサ特権レベルの観点から表現される。いくつかの実施形態において、ハイパーバイザ３０はプロセッサ１２の制御権を最高の特権レベル（典型的にはルートモードとして、またはＩｎｔｅｌプラットフォーム上のＶＭＸｒｏｏｔとして知られる）で得て、よってホストシステム１０上で実行する他のソフトウェアに仮想マシン３２として提示されるハードウェア仮想化プラットフォームを作成する。図２のＯＳ３４ａ～ｂのようなオペレーティングシステム３４がＶＭ３２の仮想環境内で実行し、ＯＳ３４はハイパーバイザ３０よりも低いプロセッサ特権（例えば、カーネルモードまたはＩｎｔｅｌプラットフォーム上のリング０）を有する。ＯＳ３４はさらにＯＳ特権レベルで実行する

保護ＯＳオブジェクト３６ａを含んでよい。いくつかの実施形態において、保護ＯＳオブジェクト３６ａは、さらに以下に示すように、ドライバオブジェクトなどのマルウェア保護するように選択されるデータ構造を備える。図２のアプリケーション４２ａ～ｄのようなアプリケーション４２ｅ～ｆの集合がＯＳ３４の特権レベルよりも低いプロセッサ特権レベル（例えば、Ｉｎｔｅｌプラットフォーム上のユーザモード）で実行する。

【００１７】

[0028]いくつかの実施形態において、マルウェア対策アプリケーション４４が典型的にはアプリケーション４２ｅ～ｆと同じプロセッサ特権レベルでＯＳ３４上で実行する。例示的なマルウェア対策アプリケーション４４には、アンチウイルスプログラム、またはアンチウイルス、アンチスパイウェアおよび他のコンピュータセキュリティアプリケーションを備えるより大きなソフトウェアスイートが含まれる。いくつかの実施形態において、マルウェア対策ドライバ３６ｂがＯＳ３４と同様のプロセッサレベルで実行する。ドライバ３６ｂは、例えばメモリのマルウェアシグネチャをスキャンするためにおよび／またはＯＳ３４上で実行するソフトウェアオブジェクトのマルウェアを示す挙動を検出するために、機能をマルウェア対策アプリケーション４４に提供する。

【００１８】

[0029]いくつかの実施形態において、イントロスペクションエンジン４０は実質的にハイパーバイザ３０と同じ特権レベルで実行し、またＶＭ３２などの仮想マシンのイントロスペクション（内観）を実施するように構成される。ＶＭの、またはそれぞれのＶＭ上で実行するソフトウェアオブジェクトのイントロスペクションはとりわけ、ソフトウェアオブジェクトの挙動を分析すること、そのようなソフトウェアオブジェクトのメモリアドレスを決定および／またはアクセスすること、特定のプロセスによるそのようなアドレスに配置されるメモリの内容へのアクセスを制限すること、およびそのような内容を分析することを含んでよい。いくつかの実施形態において、イントロスペクションエンジン４０の対象となるソフトウェアオブジェクトはプロセス、命令ストリーム、レジスタ、およびとりわけページテーブルやドライバオブジェクトなどのデータ構造を備える。

【００１９】

[0030]例示的な動作では、メモリイントロスペクションエンジン４０はマルウェアから保護されるソフトウェアオブジェクトから成る保護エリア３８を設定してよい。そのようなオブジェクトを保護することは、それぞれのオブジェクトに属するデータを記憶するメモリ領域へのアクセスを制限することを含んでよい。ハイパーバイザ３０がＶＭ３２のメモリ空間の制御権を有しているので、ＯＳ３４により使用されるメモリの特定の領域を保護することは、例えばハイパーバイザ３０がそれぞれのメモリ領域への適切なアクセス権を設定することにより達成してよい。いくつかの実施形態において、保護エリア３８は保護ＯＳオブジェクト３６ａおよびマルウェア対策ドライバ３６ｂを含む。ＯＳ３４がＬｉｎｕｘ（登録商標）オペレーティングシステムである場合、例示的な保護ＯＳオブジェクト３６ａはとりわけカーネル（ｓｙｓ＿ｃａｌｌ＿ｔａｂｌｅなどの読み取り専用コードおよび／またはデータ）、ｓｙｓｅｎｔｅｒ／ｓｙｓｃａｌｌ制御レジスタ、ならびにアドレスｉｎｔ　０ｘ８０（ｓｙｓｃａｌｌ）および／またはｉｎｔ　０ｘ０１を含む。Ｗｉｎｄｏｗｓ（登録商標）　ＯＳの例示的な保護オブジェクトはカーネル（システムサービスディスパッチテーブルを含む、読み取り専用コードおよび／またはデータ）、各種記述子テーブル（例えば、割り込み、全体および／またはローカル）、ｓｙｓｅｎｔｅｒ／ｓｙｓｃａｌｌ制御レジスタならびに／または割り込み記述子テーブルレジスタ（ＩＤＴＲ）、グローバル記述子テーブルレジスタ（ＧＤＴＲ）、およびローカル記述子テーブルレジスタ（ＬＤＴＲ）などの他のレジスタを含む。いくつかの実施形態において、保護ＯＳオブジェクト３６ａはとりわけ、特定のドライバオブジェクトおよび高速Ｉ／Ｏディスパッチテーブル（例えば、ｄｉｓｋ、ａｔａｐｉ、ｃｌｆｓ、ｆｌｔｍｇｒ、ｎｔｆｓ、ｆａｓｔｆａｔ、ｉａｓｔｏｒ、ｉａｓｔｏｒｖ）も備えてよい。他の保護ＯＳオブジェクト３６ａは、ｉａ３２＿ｓｙｓｔｅｎｔｅｒ＿ｅｉｐ、ｉａ３２＿ｓｙｓｅｎｔｅｒ＿ｅｓｐ、ｉａ３２＿ｅｆｅｒ、ｉａ３２＿ｓｔａｒ、ｉａ３２＿ｌｓｔａｒ、およびｉａ

３２＿ｇｓ＿ｂａｓｅなどの特定のモデル固有レジスタ（ＭＳＲｓ：Ｍｏｄｅｌ　Ｓｐｅｃｉｆｉｃ　Ｒｅｇｉｓｔｅｒ）を含んでよい。いくつかの実施形態において、イントロスペクションエンジン４０はまた、悪意のあるコードを収容するアドレスへの不正な再ルーティングを防止するために、ページテーブルを保護する。

【００２０】

[0031]仮想マシンは典型的には仮想化物理メモリ、すなわちホストシステム１０の実際の物理メモリ１４の仮想表現で動作する。仮想化物理メモリは各ゲストＶＭ３２ａ～ｂに固有の仮想化アドレスの連続した空間を備え、上記空間の部分が物理メモリ１４および／または物理記憶装置２０内のアドレスにマッピングされる。仮想化をサポートするように構成されたシステムでは、そのようなマッピングは典型的には拡張ページテーブル（ＥＰＴ：Ｅｘｔｅｎｄｅｄ　Ｐａｇｅ　Ｔａｂｌｅ）または入れ子ページテーブル（ＮＰＴ：Ｎｅｓｔｅｄ　Ｐａｇｅ　Ｔａｂｌｅ）などの、プロセッサ１２により制御される専用のデータ構造により達成される。そのようなシステムでは、仮想化物理メモリは当該技術分野でページとして知られる単位で分割してよく、ページは物理メモリにＥＰＴおよび／またはＮＰＴなどの機構を介して個別にマッピングされる仮想化物理メモリの最小単位を表し、すなわち物理および仮想化物理メモリ間のマッピングはページ粒度で実施される。すべてのページは典型的には所定のサイズ、例えば、４キロバイト、２メガバイトなどを有する。仮想化物理メモリの分割は通常ハイパーバイザ３０により構成される。いくつかの実施形態において、ハイパーバイザ３０はＥＰＴ／ＮＰＴおよびその結果物理メモリと仮想化物理メモリとの間のマッピングをも構成する。いくつかのハードウェア構成では、ハイパーバイザ３０は例えばそれぞれのページへの読み書きアクセス権を設定することにより、各ページ内に格納されるデータへのアクセスを選択的に制御することができるようになる。そのような権利は例えばＥＰＴまたはＮＰＴ内のそれぞれのページのエントリを改変することにより設定してよい。ハイパーバイザ３０はこのように、どのソフトウェアオブジェクトが各ページ内のアドレスに記憶されるデータにアクセスしてよいか選択してよく、また例えば、読み取り、書き込みなど、どの操作がそれぞれのデータに許可されるか示してよい。ソフトウェアオブジェクトによる、そのオブジェクトがそれぞれの権利を有していないページに対する、データを読み取る、またはデータを書き込むなどの操作を実施しようとする試行は仮想マシン終了イベント（例えばＩｎｔｅｌプラットフォーム上のＶＭＥｘｉｔイベント）をトリガしてよい。いくつかの実施形態において、仮想マシン終了イベントはプロセッサの制御権をそれぞれのソフトウェアオブジェクトを実行するＶＭからハイパーバイザ３０に移譲し、よって不正な読み／書きの試行を防止する。

【００２１】

[0032]いくつかの実施形態において、ＯＳ３４は仮想メモリ空間（論理アドレス空間とも呼ぶ）を構成し、上記仮想メモリ空間を図３でのアプリケーション４２ｅ～ｆおよび４４などのアプリケーションに、および／または保護オブジェクト３６ａ～ｂなどの別のソフトウェアオブジェクトに公開する。そのようなシステムでは、ＯＳ３４は例えばページテーブル機構を使用して、上記仮想メモリ空間とＶＭ３２の仮想化物理メモリとの間のマッピングを構成し維持する。いくつかの実施形態において、上記仮想メモリ空間はまたページに分割され、そのようなページはＯＳ３４により仮想化物理メモリに個別にマッピングされる仮想メモリの最小単位を表す（仮想から仮想化物理メモリへのマッピングはページ粒度で実施される）。

【００２２】

[0033]図４は、図２に示す実施形態におけるメモリアドレスの例示的なマッピングを例示する。アプリケーション４２ａまたはゲストＯＳ３４ａなどのソフトウェアオブジェクトにはゲストＯＳ３４ａにより仮想アドレス空間２１４ａが割り当てられる。それぞれのソフトウェアオブジェクトが例示的なメモリアドレス５０ａにアクセスしようと試行すると、アドレス５０ａはゲストＶＭ３２ａの仮想化プロセッサにより、ゲストＯＳ３４ａにより構成され制御されるページテーブルに従って、仮想マシン３２ａの仮想化物理メモリ空間１１４ａ内のアドレス５０ｂに変換される。アドレス５０ｂは当該技術分野でゲスト

物理アドレスとしても知られる。仮想化物理メモリ１１４ａを構成し制御するハイパーバイザ３０は次いで例えば上述のようにＥＰＴまたはＮＰＴ手段を使用して、アドレス５０ｂをホストシステム１０の物理メモリ１４内のアドレス５０ｃにマッピングする。

【００２３】

[0034]同様に、仮想メモリ空間２１４ｂがゲストＯＳ３４ｂによりゲストＶＭ３２ｂ上で実行するアプリケーション（例えば４２ｃ）または他のソフトウェアオブジェクトのために設定される。空間２１４ｂ内の例示的な仮想アドレス５０ｄがゲストＶＭ３２ｂの仮想化プロセッサにより、ゲストＯＳ３４ｂにより構成され制御される変換テーブルに従って、ゲストＶＭ３２ｂの仮想化物理メモリ空間１１４ｂ内のアドレス５０ｅに変換される。アドレス５０ｅはさらにハイパーバイザ３０により物理メモリ１４内のアドレス５０ｆにマッピングされる。

【００２４】

[0035]いくつかの実施形態において、ハイパーバイザ３０は物理メモリ１４の表現を備える自身の仮想メモリ空間２１４ｃを設定し、空間２１４ｃでのアドレスを物理メモリ１４でのアドレスにマッピングするために変換機構（例えば、ページテーブル）を活用する。図４では、そのような例示的なマッピングはアドレス５０ｇをアドレス５０ｈに変換する。同様に、物理メモリ１４での５０ｃと５０ｆなどのアドレスはハイパーバイザ３０の仮想メモリ空間２１４ｃ内でそれぞれアドレス５０ｋと５０ｍに対応する。

【００２５】

[0036]ＯＳ３４ａ～ｂの本質的なタスクは、メモリの部分をそれぞれのＶＭ３２ａ～ｂ上で実行するソフトウェアオブジェクトに動的に割り当てることと、そのような部分がもはやそれぞれのプロセスにより必要とされない場合に、それらを再利用できるように解放することである。そのようなメモリ割り当ては仮想メモリ２１４ａ～ｂのレベルで、またはそれぞれの仮想マシンの仮想化物理メモリ１１４ａ～ｂのレベルで実行してよい。メモリ割り当ておよび割り当て解除は典型的には、Ｗｉｎｄｏｗｓ（登録商標）でのＫｅＡｌｌｏｃａｔｅＰｏｏｌＷｉｔｈＴａｇやＫｅＦｒｅｅＰｏｏｌＷｉｔｈＴａｇなどの専用のＯＳメモリ管理関数により実施される。

【００２６】

[0037]前述のメモリマッピング機構により、対象オブジェクトに対するメモリ割り当ては常に、それぞれのＶＭの仮想化物理メモリの一部分を対象オブジェクトに割り当てる結果になる。それぞれの部分のサイズは、ページサイズとそれぞれのソフトウェアオブジェクトのメモリ要件とに従って、一ページの一部分のみ、または一ページを超えてよい。以下では一部分は、ページの集合の各ページがそれぞれの部分の一部を含む場合、ページの集合に及ぶと言う。いくつかのオブジェクトに同じページの別個の部分が割り当てられてよい。図５は、複数のページ２６ａ～ｄに分割される仮想化物理メモリの例示的な領域、および各々別個のソフトウェアオブジェクトに割り当てられる２つの部分５２および５４が同じページ２６ｂに広がる例示的なメモリ割り当てを示す。別の例示的な部分５６は２つのページ２６ｃ～ｄに及ぶ。

【００２７】

[0038]対照的に図６は、本発明のいくつかの実施形態に係る例示的なメモリ割り当てを示す。以下に詳細に示すように、対象オブジェクトがマルウェア保護を必要とする場合、例えば対象オブジェクトが保護エリア３８に属するドライバオブジェクトまたは他のオブジェクトである場合、保護準備モジュール４６のいくつかの実施形態は対象オブジェクトに対するメモリ割り当てを改変し、その結果対象オブジェクトの一部を含む各ページは対象オブジェクトのために予約される。

【００２８】

[0039]当業者は、ページの集合を対象オブジェクトのために予約することはいくつかの方法で達成してよいことを理解するであろう。いくつかの実施形態において、ページを予約することは、部分５８がページ２６ｆ全体を占める図６の例に示すように、ページ全体を対象オブジェクトに割り当てることを含む。そのような排他的な割り当ては例えば、以

下により詳細に示すように、それぞれのオブジェクトに割り当てられる部分のサイズをページサイズの整数倍に等しい新たなサイズに変更することにより達成してよい。対照的に図６の同じ例で、無保護ソフトウェアオブジェクトには別のページ２６ｇのほんの一部分から成る部分６０が割り当てられる。他の実施形態において、ページの集合を対象オブジェクトのために予約することは、必ずしも対象オブジェクトのサイズを改変することまたはページの集合全体を対象オブジェクトに割り当てることを含まないが、そうでなければ対象オブジェクト以外のオブジェクトにはそれぞれのページの集合内のメモリ空間は一切割り当てられないことを保証することにより達成してよい。例えば、ページを対象オブジェクトのために予約することは、以下により詳細に示すように、例えばＯＳ３４の初期化時に空ページの集合を予約すること、およびオブジェクトをそれぞれの空ページの集合内に割り当てる（または移動する）ことを含んでよい。

【００２９】

[0040]図７は、本発明のいくつかの実施形態に係るメモリイントロスペクションエンジン４０により実施されるステップの例示的なシーケンスを示す。エンジン４０はホストシステム１０上で同時に実行する複数の仮想マシンを検査および／または保護するように構成してよい。図７に例示するステップはそのような各仮想マシンに対して繰り返してよい。ステップ１０２で、エンジン４０はＯＳ３４の初期化を検出する。いくつかの実施形態において、ステップ１０２はハイパーバイザ３０のレベルから、ＯＳ初期化を示すプロセッサイベントをリッスンすることを含む。ＯＳ初期化動作を実施するプロセッサ命令は典型的には、例えばＩｎｔｅｌプラットフォーム上のリング０など、カーネルプロセッサ特権を必要とする。ＯＳ３４の特権レベルから実行されたとき、そのような命令は、プロセッサ１２の制御権をＯＳ３４からハイパーバイザ３０に移譲する、Ｉｎｔｅｌプラットフォーム上のＶＭＥｘｉｔイベントなどの仮想マシン終了イベントをトリガしてよい。仮想マシン終了イベントはしたがってハイパーバイザ３０および／またはイントロスペクションエンジン４０により分析してよい。ＯＳ初期化を示すイベントおよび／またはプロセッサ命令の例には割り込みおよびグローバル記述子テーブルの初期化が含まれ、これらはＷｉｎｄｏｗｓ（登録商標）　ＯＳの初期化の早い段階でＬＩＤＴおよび／またはＬＧＤＴなどの特権命令を使用することにより行われる。ＯＳ初期化を示すイベント／命令の他の例には、ＷＲＭＳＲマシン命令を使用することによりＳＹＳＥＮＴＥＲおよび／またはＳＹＳＣＡＬＬモデル固有レジスタに書き込むことが含まれ、これはＯＳのカーネルイメージがメモリにロードされた直後に行われる。さらに他の例には、とりわけマシンチェック構成レジスタ、およびＦＳ／ＧＳベースレジスタなどの、ＯＳ３４によりプログラムされる他のモデル固有レジスタの初期化が含まれる。いくつかの実施形態おいて、そのようなイベント／命令を検出することはＯＳ３４が初期化されたことを示す。

【００３０】

[0041]ステップ１０４で、メモリイントロスペクションエンジン４０のいくつかの実施形態はそれぞれのゲストＶＭ上で現在実行中または初期化中のオペレーティングシステムの種類を識別する。例示的なＯＳ種類にはとりわけＷｉｎｄｏｗｓ（登録商標）、Ｌｉｎｕｘ（登録商標）、ＭａｃＯＳ、およびＡｎｄｒｏｉｄが含まれる。ＯＳ種類はＷｉｎｄｏｗｓ（登録商標）などの名称標識、およびとりわけ７、ＨｏｍｅまたはＥｎｔｅｒｐｒｉｓｅなどのバージョン標識を備えてよい。いくつかの実施形態において、ステップ１０４はそれぞれのゲストＶＭのモデル固有レジスタ（ＭＳＲ）の内容に、またはそれぞれのＭＳＲにより指し示されるメモリの一部分の内容に従ってＯＳの種類を識別することを含む。いくつかの実施形態において、エンジン４０はそのようなＭＳＲにステップ１０２で遮断された命令により書き込まれるデータに従ってＯＳの名称を決定してよい。例えばエンジン４０は、ＳＹＳＥＮＴＥＲにまたはＳＹＳＣＡＬＬ　ＭＳＲに書き込む命令を遮断し、現在実行中、または現在初期化中のＯＳの種類をそのような書き込み命令のパラメータに従って決定してよい。ＯＳ名称に関する情報を提供してよい他の例示的なレジスタには、とりわけ制御レジスタ、割り込み記述子テーブル（ＩＤＴ）、およびグローバル記述子テーブル（ＧＤＴ）が含まれる。ＭＳＲ書き込みに従ってＯＳ種類を識別するために、

イントロスペクションエンジン４０はさらに各ＯＳに固有の高速システムコールハンドラ（例えば、ＳＹＳＣＡＬＬまたはＳＹＳＥＮＴＥＲ　ＭＳＲの内容に従って処理されるシステムコール）の所定のライブラリに対するパターンマッチングを使用してよい。そのような高速システムコールライブラリはメモリイントロスペクションエンジン４０を設けてよく、また定期的またはオンデマンドのソフトウェア更新を介して最新の状態に保ってよい。

【００３１】

[0042]いくつかの実施形態において、バージョン標識（例えばリリース名称、ビルド番号など）はＯＳのそれぞれの種類に固有の特定のカーネルデータ構造を構文解析することにより得てよい。ＯＳバージョンの識別を可能にする例示的なデータ構造はＬｉｎｕｘ（登録商標）カーネルの特定のエクスポートシンボルまたは、とりわけＮｔＢｕｉｌｄＮｕｍｂｅｒなどの、Ｗｉｎｄｏｗｓ（登録商標）カーネルの特定のエクスポートされるシンボルである。

【００３２】

[0043]ステップ１０６で、メモリイントロスペクションエンジン４０は、エンジン４０が完全性保護を適用することに備えて、保護準備モジュール４６にメモリ準備動作を実施するよう要求してよい。いくつかの実施形態において、そのようなメモリ準備動作はマルウェア保護するように選択される対象オブジェクトの割り当てを改変して、対象オブジェクトの一部を含むすべてのページがそれぞれの対象オブジェクトのために予約されるようにすることを含む。いくつかの実施形態において、ステップ１０６はマルウェア保護するように選択されるオブジェクトに後に排他的に割り当てられる未割り当てメモリページの予約プールを確立することを含んでよい。そのような予約プールを確立するために、保護準備モジュール４６はそれぞれのプールを所定のサイズ（例えば、２０ＭＢ）のダミーソフトウェアオブジェクトに割り当てるためにネイティブのＯＳメモリ割り当て関数を呼び出してよい。モジュール４６はこのようにそれぞれのメモリのプールはネイティブのＯＳメモリ管理関数により実施されるさらなる割り当てには使用されないことを保証してよい。モジュール４６の機能は図８～１０に関連してさらに以下に詳述する。

【００３３】

[0044]ステップ１０８で、メモリイントロスペクションエンジン４０は保護する対象オブジェクトを選択する。いくつかの実施形態において、ステップ１０８はマルウェア保護を必要とするソフトウェアオブジェクト（図３での保護エリア３８を参照）を、ステップ１０２～１０４で決定されたＯＳの種類に従って識別することを含む。保護の対象となる例示的なオブジェクトはとりわけＯＳドライバオブジェクトとマルウェア対策ドライバ３６ｂである。そのようなオブジェクトを識別するために、メモリイントロスペクションエンジン４０はマルウェア保護を必要とするＯＳ固有オブジェクトのリストを維持してよく、リストは定期的および／またはオンデマンドのソフトウェア更新により最新の状態に保ってよい。いくつかの実施形態において、エンジン４０はＯＳ３４によるドライバをロードしようとする各試行を遮断し、それぞれのドライバの完全性および／またはシグネチャチェックのセットを実施してよい。そのようなチェックはそれぞれのドライバの一部のハッシュを既知のドライバに対して決定されたハッシュのライブラリとマッチングすることを含んでよく、またＯＳ３４が使用中および／または現在ロードしようと試行中の複数のドライバのうちの、マルウェア対策ドライバ３６ｂの識別を可能にしてよい。一旦識別されると、ドライバ３６ｂは、以下により詳細に示すように、ＯＳの他のコンポーネントと共に保護してよい。

【００３４】

[0045]ステップ１１０は対象オブジェクトが初期化されるまで待機する。例えばドライバが初期化されると、ＯＳ３４はいくつかの正規の構成書き込みを、それぞれのドライバに、またはそれぞれのドライバに従ってもしくはそれぞれのドライバが制御するように構成された装置に従って選択される他のソフトウェアオブジェクトに割り当てられるメモリ空間に実施し、エンジン４０はそのような正規の書き込みがステップ１１０の一部として

進行できるようにしてよい。対象オブジェクトが初期化されたかどうかを判定するために、エンジン４０のいくつかの実施形態はドライバ初期化を示すイベントおよび／またはプロセッサ命令をリッスンしてよい。そのようなイベントの例には、とりわけそれぞれのオブジェクトの特定のエリアへの改変が含まれる。

【００３５】

[0046]ステップ１１２で、エンジン４０はそれぞれの対象オブジェクトを、例えばＯＳ３４を侵害しようと試行する悪意のあるソフトウェアによる不要な改変から保護する。いくつかのそのようなメモリ保護機構が当該技術分野で公知である。いくつかの実施形態において、対象オブジェクトを保護することは、ＯＳ３４によりそれぞれの対象オブジェクトに割り当てられるメモリ空間を書き込み保護することを含む。そのような書き込み保護は、メモリイントロスペクションエンジン４０の要求に応じてハイパーバイザ３０により、ＥＰＴまたはＮＰＴなどのデータ構造を使用して実施してよい。例えばハイパーバイザ３０は対象オブジェクトに割り当てられるメモリページを、それぞれのページのＥＰＴ／ＮＰＴアクセス権ビットを改変することにより、読み取り専用として設定してよい。あるいはハイパーバイザ３０は対象オブジェクトに割り当てられるメモリページへ書き込みしようとする任意の試行を遮断し、それぞれの試行を分析のためにメモリイントロスペクションエンジン４０にリダイレクトしてよい。書き込み試行を分析した後、メモリイントロスペクションエンジン４０はそれぞれの書き込み動作を許可するかまたは拒否（中止）するかを決定してよい。ステップ１０８～１１２はマルウェア保護を必要とするすべての対象オブジェクトに対して繰り返してよい。

【００３６】

[0047]いくつかの実施形態において、保護準備モジュール４６により実施されるメモリ準備動作はマルウェア保護するように選択される対象ソフトウェアオブジェクトのメモリ割り当てを改変することを含み、上記の改変は前段で示したように（図７、ステップ１０８～１１２）、エンジン４０がそれぞれのソフトウェアオブジェクトの完全性保護を適用する以前に起きる。いくつかの実施形態において、モジュール４６は、対象オブジェクトの一部を含むメモリページがそれぞれの対象オブジェクトのために予約されるように、メモリ割り当てを改変する。上記のメモリ割り当てを改変することは、ＯＳ３４にネイティブのメモリ管理関数を実行した結果を改変すること、および／またはそれぞれのメモリ管理関数自体を改変することを含んでよい。ネイティブのメモリ管理関数はＯＳ３４の製造者により提供されるソフトウェアオブジェクトを備え、ネイティブの関数はメモリ割り当ておよび割り当て解除動作を実施する。そのような関数の例はＷｉｎｄｏｗｓ（登録商標）　ＯＳにネイティブのＫｅＡｌｌｏｃａｔｅＰｏｏｌＷｉｔｈＴａｇやＫｅＦｒｅｅＰｏｏｌＷｉｔｈＴａｇである。

【００３７】

[0048]いくつかの実施形態において、保護準備モジュール４６はステップ１０４でイントロスペクションエンジンにより決定されたＯＳ種類に従って、ネイティブのメモリ管理関数の集合を、例えばそのような関数がそれぞれのゲスト仮想マシン３２のメモリに常駐するメモリアドレスを決定することにより識別する。そのようなメモリアドレスを決定するために、モジュール４６はカーネルバイナリイメージ（例えばＷｉｎｄｏｗｓ（登録商標）でのＰｏｒｔａｂｌｅ　Ｅｘｅｃｕｔａｂｌｅ、Ｌｉｎｕｘ（登録商標）でのＥｘｅｃｕｔａｂｌｅ　ａｎｄ　Ｌｉｎｋａｂｌｅ　Ｆｏｒｍａｔ）のエクスポートされる関数表などの特定のデータ構造にアクセスしてよい。

【００３８】

[0049]ネイティブのメモリ管理関数の識別後、モジュール４６のいくつかの実施形態は続いて上記の関数を追加機能を提供することにより改変してよい。そのような改変は当該技術分野で公知の任意のフッキング方法を使用して達成してよい。例えばモジュール４６は、それぞれのネイティブの関数に上書きまたは追加される、ＶＭＣａｌｌ命令またはＪＭＰ命令などの、リダイレクションパッチを適用してよい。他の実施形態はそれぞれのメモリ管理関数のＥＰＴエントリを、新たなアドレスを指し示すように、改変してよい。い

くつかの実施形態において、そのようなパッチおよび／またはＥＰＴフックの効果は、ネイティブの関数の実行を保護準備モジュール４６により提供されるコードの断片にリダイレクトすることであり、そのようなコードの例示的な機能は以下に提示する。フッキング後、ＯＳ３４が対象オブジェクトに関してメモリを割り当てようとまたはメモリを割り当て解除しようと試みると、コードの断片がそれぞれのネイティブのＯＳメモリ管理関数のコードの前にまたは代わりに実行されるであろう。

【００３９】

[0050]図８は、本発明のいくつかの実施形態に係る、Ｗｉｎｄｏｗｓ（登録商標）でのＫｅＡｌｌｏｃａｔｅＰｏｏｌＷｉｔｈＴａｇなどのネイティブのＯＳメモリ割り当て関数の改変を含むステップの例示的なシーケンスを示す。シーケンスは保護準備モジュール４６の一部を形成してよく、またそれぞれのメモリ割り当て関数に適用されるパッチ／フックにより実施されるリダイレクションの結果として実行してよい。ステップ１２２は、メモリ割り当てを要求するオブジェクトがマルウェア保護の資格があるかどうかを判定するために選択基準を適用する。そのような判定は例えばそれぞれの関数呼び出しのパラメータおよび／または引数に従って行われてよい。例示的な選択基準は、例えばドライバオブジェクトなど、その種類に従ってオブジェクトを選択することを含む。Ｗｉｎｄｏｗｓ（登録商標）系統からＯＳ３４を実行する実施形態において、割り当て中のオブジェクトの種類はネイティブのメモリ割り当て関数ＫｅＡｌｌｏｃａｔｅＰｏｏｌＷｉｔｈＴａｇの割り当てタグに従って決定してよい。例えば「Ｄｒｉｖ」タグはドライバオブジェクトを示す。代替の選択基準は、割り当てを要求するオブジェクトがマルウェア保護の対象となるオブジェクトのリストにあるかどうかを判定することを含む。そのようなリストは保護エリア３８（図３）の構成要素を含んでよい。いくつかの実施形態において、追加の選択基準は、それぞれのオブジェクトのメモリ割り当てが後述の方法により安全に改変できるかどうかを判定することを含む。

【００４０】

[0051]ステップ１２４は対象オブジェクトがマルウェア保護の選択基準を満たすかどうかを判定する。いいえの場合、ステップ１２８は実行の制御権をそれぞれのＯＳのネイティブのメモリ割り当て関数に返す。対象オブジェクトが保護するように選択される場合、ステップ１２６で、保護準備モジュール４６のいくつかの実施形態は対象オブジェクトのサイズをページサイズの整数倍に等しい新たなオブジェクトサイズに変更する。そのようにしてオブジェクトサイズを変更することは、効果的にメモリアロケータに、オブジェクトを収容するために必要な実際のメモリの量の代わりにページ全体の集合をそれぞれの対象オブジェクトに割り当てさせるかもしれない。いくつかの実施形態において、対象オブジェクトのサイズはページサイズの次に最も近い整数倍に丸められる。例えば３２０バイトの対象オブジェクトには４ｋＢのページ全体を割り当ててよく、６ｋＢのオブジェクトには２つの４ｋＢのページ全体を割り当ててよい。いくつかの実施形態において、例えばサイズが少なくとも一ページであるメモリの一部分を割り当てる、Ｗｉｎｄｏｗｓ（登録商標）　ＯＳを実行する仮想マシンは、割り当てられる部分をページ境界に自動的に揃える（例えば、図６での部分５８）。他の実施形態において、ステップ１２６は割り当てられる部分をページ境界に明示的に揃えることを含んでよく、その結果部分が及ぶすべてのページが対象オブジェクトに排他的に割り当てられる。当業者は、部分をページ境界に揃えることを達成する方法が多く存在し得ることを理解するであろう。例えばステップ１２６は、対象オブジェクトのサイズを一ページのサイズで大きくすること、およびポインタを結果的に割り当てられるメモリの部分へ変更することを含んでよい。ステップ１２６を完了した後、実行の制御権はネイティブのＯＳメモリ管理関数に返される。

【００４１】

[0052]図９は、保護ソフトウェアオブジェクトに割り当てられる部分が及ぶすべてのページがそれぞれのオブジェクトのために予約されるメモリ割り当てを保護準備モジュール４６が達成してよい代替の様式を例示するステップの例示的なシーケンスを示す。図９に示すシーケンスは、Ｗｉｎｄｏｗｓ（登録商標）でのＫｅＡｌｌｏｃａｔｅＰｏｏｌＷｉ

ｔｈＴａｇなどのネイティブのＯＳメモリ割り当て関数の機能を改変することを含み、またそれぞれのメモリ割り当て関数に適用されるパッチ／フックに応答して実行してよい。ステップ１３２～１３４のシーケンスは対象オブジェクトがマルウェア保護の基準を満たすかどうかを検証するものであり、ステップ１３２～１３４は上述のステップ１２２～１２４と同様の様式で進めてよい。対象オブジェクトが保護するようには選択されない場合、ステップ１３８はプロセッサの制御権をそれぞれのネイティブのメモリ割り当て関数に返す。対象オブジェクトがマルウェア保護するように選択される場合、ステップ１３６はネイティブの割り当て関数をバイパスし、保護オブジェクトのために予約されるメモリの領域内に配置される一部分を直接割り当てる。いくつかの実施形態において、ステップ１３６はＯＳ３４の初期化時にモジュール４６により確立されるメモリの予約プール内に配置されるアドレスを示す割り当てポインタを決定することを含む（図７、ステップ１０６に関連して上記参照）。いくつかの実施形態において、モジュール４６はさらに割り当てられる部分がページ境界に揃うように割り当てポインタを決定してよい。次にステップ１４０はステップ１３６で決定された割り当てのポインタを返す。

【００４２】

[0053]いくつかの実施形態において、保護準備モジュール４６はまた、ネイティブのＯＳメモリ割り当て解除関数を実行した結果を変更することにより、またはそれぞれの関数自体を改変することにより割り当て解除プロセスを改変してよい。そのような改変は例えば、ＫｅＦｒｅｅＰｏｏｌＷｉｔｈＴａｇなどのネイティブのＯＳメモリ割り当て解除関数をフックして追加の機能を含めることにより達成してよい。図１０は、本発明のいくつかの実施形態に係る、メモリ割り当て解除関数のそのような改変を含むステップの例示的なシーケンスを示す。ステップ１４２～１４４のシーケンスは割り当て解除中のオブジェクトのアドレスがイントロスペクションエンジン４０および／またはハイパーバイザ３０により書き込み保護されている（上記参照）かどうかを判定する。それぞれのアドレス／ページがそのような保護を有していない場合、ステップ１４８は実行の制御権をネイティブのＯＳ割り当て解除関数に返す。それぞれのアドレス／ページがエンジン４０および／またはハイパーバイザ３０により書き込み保護されている場合、ステップ１４６で、メモリイントロスペクションエンジン４０および／またはハイパーバイザ３０の構成要素はステップ１４８に進む前に保護を解除してよい。いくつかの実施形態において、それぞれのオブジェクトの保護を解除することは、ＥＰＴまたはＮＰＴの変化を操作してそれぞれのオブジェクトのアドレスを収容するページはもはや、例えば読み取り専用など、書き込み保護されていないことを示すことを含む。

【００４３】

[0054]上述の例示的なシステムおよび方法はコンピュータシステムなどのホストシステムをウイルスやルートキットなどのマルウェアから保護することを可能にする。従来のシステムでは、ルートキットはオペレーティングシステムと実質的に同様のプロセッサ特権レベルで動作することによりコンピュータシステムの制御権を得るかもしれない。対照的に本発明のいくつかの実施形態において、ハイパーバイザは最高特権レベルでコンピュータシステム上で実行し、オペレーティングシステムを仮想マシンに置き換える。いくつかの実施形態において、メモリイントロスペクションエンジンはハイパーバイザと同じプロセッサ特権レベルで実行する。マルウェア対策動作はしたがってオペレーティングシステムよりも高いプロセッサ特権レベルから行ってよい。いくつかの実施形態において、単一のメモリイントロスペクションエンジンはそれぞれのコンピュータシステム上で同時に実行する複数の仮想マシンを保護してよい。

【００４４】

[0055]いくつかの実施形態において、それぞれのシステムをマルウェアから保護することは、とりわけ特定のドライバ、レジスタ、およびページテーブルなどの重要なソフトウェアオブジェクトの集合を選択すること、およびそのようなオブジェクトへの悪意のある変更を防止することを含む。対象オブジェクトを保護するために、いくつかの実施形態はそれぞれのオブジェクトに割り当てられるメモリ空間へ書き込みしようとする試行を遮断

することにより、そのような悪意のある変更を防止してよい。そのような遮断はハイパーバイザのレベルから実施してよい。

【００４５】

[0056]他の実施形態はそれぞれのオブジェクトに割り当てられるメモリ空間を読み取り専用として記すことにより対象オブジェクトを保護してよい。典型的なハードウェアおよびソフトウェア構成では、メモリはページとして知られる連続したアドレスの個々のブロックに分割される。アクセス許可、例えば読み書き許可は、典型的にはページ粒度で設定される、すなわち一ページ内のすべてのアドレスが同じアクセス許可を有する。したがって、対象オブジェクトのメモリ空間を保護することは、例えばそれぞれのオブジェクトに属するデータを含むページの集合を読み取り専用として記すことにより達成してよい。ページレベルアクセス許可は例えば、Ｉｎｔｅｌプラットフォーム上での拡張ページテーブル（ＥＰＴ）などの専用データ構造を使用して、ハイパーバイザにより制御される。

【００４６】

[0057]従来のシステムでは、ＯＳがメモリ空間をそれぞれのシステム上で実行するソフトウェアオブジェクトに割り当てる場合、複数のソフトウェアオブジェクトは、例えばそれぞれのオブジェクトが小さなメモリ使用量を有すれば、同じページ内に割り当ててよい。いくつかの状況では、それぞれのオブジェクトの１つはＯＳに重要であり、したがってマルウェアからの保護を必要とするかもしれず、一方同じページからの別のオブジェクトは頻繁に正規の書き換えを必要とするかもしれない。アクセス許可はページ粒度で設定されるので、重要なオブジェクトを保護することは、重要なオブジェクトと同じページ内に存在するすべてのオブジェクトへの書き込みアクセスを拒否する結果になるかもしれない。保護ページ内に存在するメモリアドレスへ書き込みしようとする各試行は典型的には障害に終わり、次いでプロセッサの制御権をそれぞれの仮想マシンのＯＳからハイパーバイザに移譲する仮想マシン終了イベントに続く。そのようなイベントは、それぞれの仮想マシンの状態をプロセッサへまたはプロセッサからロードおよび／またはアンロードすることを含むかもしれず、さらに計算オーバーヘッドを増す。とりわけ保護オブジェクトを収容するページに書き込みしようとする各正規の試行はしたがって、コンピュータシステムの著しい減速をもたらすかもしれない。

【００４７】

[0058]対照的に、本発明のいくつかの実施形態は、メモリがマルウェア保護を必要とするソフトウェアオブジェクトに割り当てられる様式を改変し、その結果それぞれのオブジェクトの一部を含む各メモリページはそれぞれのオブジェクトのために予約される。そのような改変は完全性保護をそれぞれのソフトウェアオブジェクトに適用する以前に実施される。対象オブジェクトとメモリページを共有する他のオブジェクトはないことを保証することにより、本発明のいくつかの実施形態は、正規の書き込み試行の遮断により生じる計算オーバーヘッドを回避しつつ、完全性保護をページ粒度で可能にする。

【００４８】

[0059]いくつかの実施形態において、それぞれのＯＳにネイティブの割り当て様式を改変することは、ＯＳのメモリ割り当ておよび割り当て解除関数を識別すること、およびそれぞれの関数をフックしてそれらの実行をハイパーバイザのプロセッサ特権レベルから実行される命令の集合にリダイレクトすることを含む。あるいは上記の改変は、それぞれの関数をそれぞれのＯＳを実行するＶＭの内で実行される命令の集合にリダイレクトし、命令の集合はハイパーバイザのレベルからそれぞれのＶＭのメモリに注入される。いくつかの実施形態において、メモリ割り当て関数は、保護を必要とするソフトウェアオブジェクトへのメモリページ全体の割り当てを強制するように改変され、割り当てられる部分はページ境界に揃えられる。したがって、保護を必要とするオブジェクトはもはや保護を必要としないオブジェクトと同じメモリページに割り当てられずにすむ。

【００４９】

[0060]代替の実施形態は、例えばＯＳ初期化時にメモリの予約プールを予約することにより、対象オブジェクトに割り当てられる各ページが対象オブジェクトのために予約され

ることを保証する。オペレーティングシステムが対象オブジェクトに対するメモリ割り当てを要求すると、いくつかの実施形態は割り当てポインタを予約プール内のアドレスにリダイレクトしてよく、こうして対象オブジェクトをページの予約集合に効果的に移動する。

【００５０】

[0061]従来のマルウェア対策解決策は典型的には単一のオペレーティングシステムに合わせて調整される。１つのオペレーティングシステムと別のとの間の切り替えには、異なるバージョンのマルウェア対策ソフトウェアをロードする必要があるかもしれない。対照的に本発明のいくつかの実施形態において、現在実行中のオペレーティングシステムの種類やバージョンにかかわらず、同じメモリイントロスペクションエンジンがそれぞれのコンピュータシステムをマルウェア保護してよい。メモリイントロスペクションエンジンをハイパーバイザのレベルで実行することにより、かつオペレーティングシステムを仮想マシンに置き換えることにより、いくつかの実施形態はいくつかのオペレーティングシステムを同時に実行し保護してよい。いくつかの実施形態において、メモリイントロスペクションエンジンは各オペレーティングシステムを例えば起動時に動的に識別してよく、さらにＯＳ固有ソフトウェアオブジェクトおよびデータ構造を保護してよい。

【００５１】

[0062]上述の実施形態は本発明の範囲から逸脱することなく多くの方法で変更してよいことは当業者には明らかであろう。したがって、本発明の範囲は以下の特許請求の範囲およびその法的均等物により定められるべきである。

【図１】

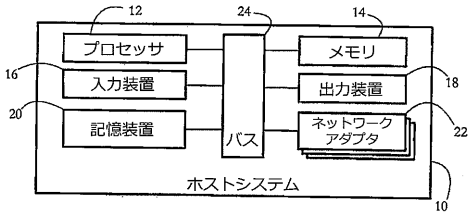

FIG. 1

【図２】

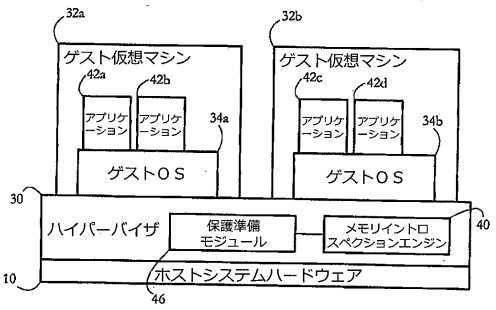

FIG. 2

【図３】

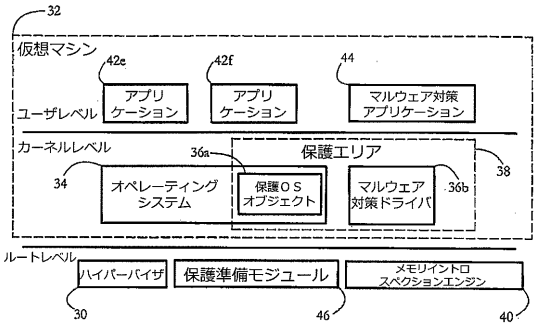

FIG. 3

【図４】

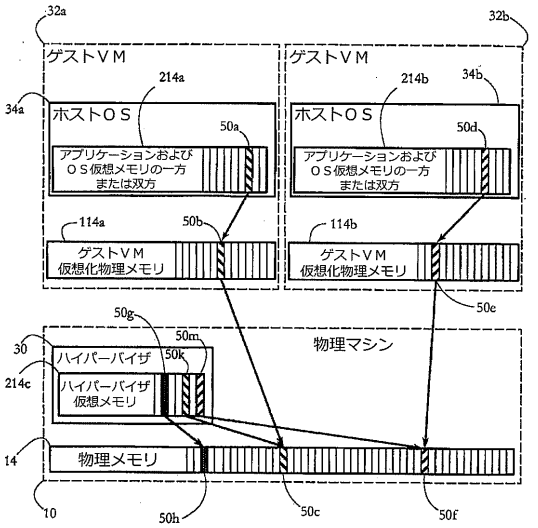

FIG. 4

【図５】

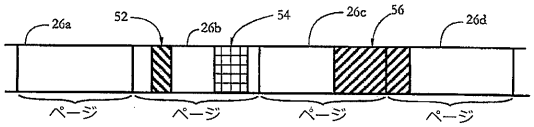

FIG. 5

【図６】

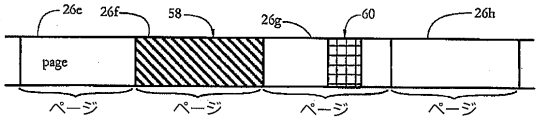

FIG. 6

【図７】

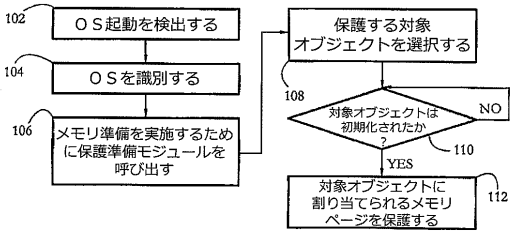

FIG. 7

【図８】

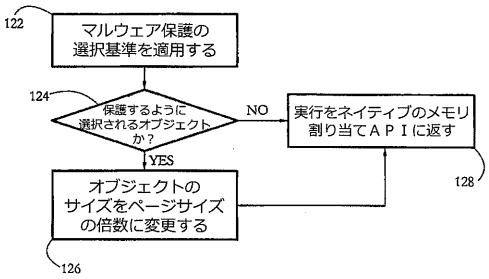

FIG. 8

【図９】

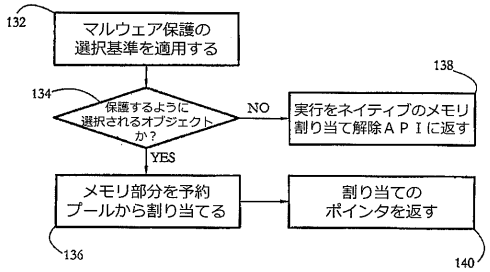

FIG. 9

【図１０】

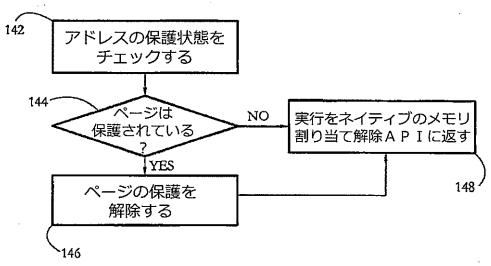

FIG. 10

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/RO2014/000006

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F12/10 G06F9/455 G06F12/14 G06F21/53 G06F21/62
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2007/055837 A1 (RAJAGOPAL PRIYA [US] ET AL) 8 March 2007 (2007-03-08) paragraphs [0005], [0007], [0014] - [0037], [0038] - [0044]; claims 1-5; figures 1, 2, -----	1-29
A	US 2009/307705 A1 (BOGNER ETAY [IL]) 10 December 2009 (2009-12-10) paragraphs [0007] - [0008], [0029] - [0049]; claim 1; figures 1, 2 -----	1-29
A	US 2012/254864 A1 (BORK JON E [US] ET AL) 4 October 2012 (2012-10-04) paragraphs [0024] - [0030]; figures 1, 6 paragraph [0041]; figure 3 paragraphs [0036] - [0048]; figures 3-4 ----- -/--	1-29

[X] Further documents are listed in the continuation of Box C. [X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 June 2014	01/07/2014

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Jardon, Stéphan

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/RO2014/000006

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2012/254993 A1 (SALLAM AHMED SAID [US]) 4 October 2012 (2012-10-04) paragraphs [0024], [0039], [0047] - [0049], [0057] - [0058], [0091], [0092]; claim 1; figures 1, 2, 3 -----	1-29
A	US 2012/331465 A1 (TANIKAWA TADAO [JP]) 27 December 2012 (2012-12-27) paragraphs [0016], [0039], [0079] - [0092], [0111], [0117], [0130] - [153192]; figures 6, 10, 11 -----	1-29

1

Form PCT/ISA/210 (continuation of second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/RO2014/000006

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2007055837 A1	08-03-2007	NONE	
US 2009307705 A1	10-12-2009	EP 2286333 A1	23-02-2011
		US 2009307705 A1	10-12-2009
		WO 2009147631 A1	10-12-2009
US 2012254864 A1	04-10-2012	NONE	
US 2012254993 A1	04-10-2012	NONE	
US 2012331465 A1	27-12-2012	CN 102859502 A	02-01-2013
		US 2012331465 A1	27-12-2012
		WO 2012117465 A1	07-09-2012

Form PCT/ISA/210 (patent family annex) (April 2005)

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CZ,DE,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IR,IS,JP,KE,KG,KN,KP,KR,KZ,LA,LC,LK,LR,LS,LT,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,SM,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US

（特許庁注：以下のものは登録商標）
１．ＡＮＤＲＯＩＤ

(72)発明者 ルツァス，アンドレイ－ヴラド
ルーマニア国ジュデツ・サトゥ・マーレ，サトゥ・マーレ，ビルド．クロシカ ヌマルル １１１
(72)発明者 ルカクス，サンドル
ルーマニア国ジュデツ・クルジュ，サト・フロレシチ，ビルド．チェタテア・フェテイ ブロック８，エタジュル ３
(72)発明者 ルツァシ，ダン－ホレア
ルーマニア国ジュデツ・クルジュ，クルジュ－ナポカ，ストラダ・ホレア ヌマルル ８９－９５，アパルタメントゥル ２９