(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2018-36695
(P2018-36695A)
(43)公開日　平成30年3月8日(2018.3.8)

(51)Int.Cl.	F I	テーマコード（参考）
G06F 21/52 (2013.01)	G06F 21/52	

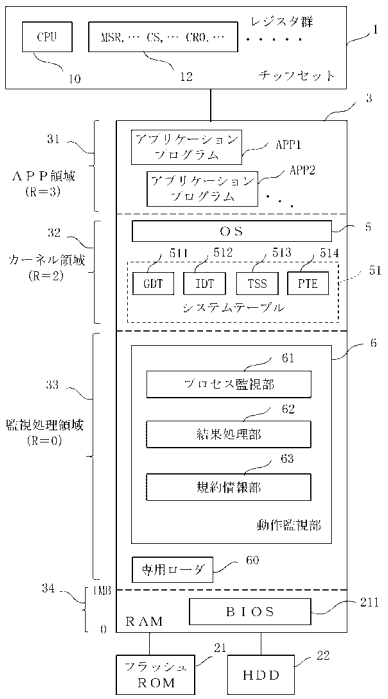
審査請求 未請求　請求項の数 9　ＯＬ　（全 11 頁）

(21)出願番号　特願2016-166689 (P2016-166689)
(22)出願日　平成28年8月29日 (2016.8.29)

(71)出願人　594170532
杉中　順子
東京都港区虎ノ門３丁目１０番４号虎ノ門ガーデン４０８号
(74)代理人　110000970
特許業務法人　楓国際特許事務所
(72)発明者　杉中順子
東京都港区虎ノ門３丁目１０番４号虎ノ門ガーデン４０８号

(54)【発明の名称】情報処理監視装置、情報処理監視方法、監視プログラム、記録媒体及び情報処理装置

(57)【要約】
【課題】ＣＰＵモードのリアルモードへの切替えを無効にすることで、リアルモードを利用することによる不当な情報処理を未然に抑止する。
【解決手段】情報処理監視装置６は、ＣＰＵモードをリアルモードに切替えるためのアクセスの発行を検出するプロセス監視部６１と、ＯＳのブート後に検出した前記アクセスに対して、アクセス動作を無効にする結果処理部６２とを備えている。
【選択図】図２



【特許請求の範囲】
【請求項１】
ＣＰＵモードをリアルモードに切替えるためのアクセスの発行を検出する監視手段と、
前記ＯＳのブート後に検出した前記アクセスに対して、アクセス動作を無効にする無効処理手段とを備えた情報処理監視装置。
【請求項２】
監視モジュールをＯＳよりも高い特権レベルで主メモリにロードする監視モジュールロード手段を備え、
前記監視モジュールは、前記監視手段及び前記無効処理手段を含むことを特徴とする請求項１記載の情報処理監視装置。
【請求項３】
前記アクセスは、ＣＰＵモードをプロテクトモードからリアルモードへ切替える実行コードである請求項１又は２に記載の情報処理監視装置。
【請求項４】
前記監視手段は、ＣＰＵモードが６４ビットモードから３２ビットのプロテクトモードモードに切替えられた状態で、前記プロテクトモードから前記リアルモードへ切替える実行コードの発行を検出することを特徴とする請求項３に記載の情報処理監視装置。
【請求項５】
前記アクセスは、ページテーブルを無効にする実行コードである請求項４に記載の情報処理監視装置。
【請求項６】
ＣＰＵモードをリアルモードに切替えるためのアクセスの発行を検出する監視手段、
ＯＳのブート後に検出した前記アクセスに対して、アクセス動作を無効にする無効処理手段、として情報処理装置を機能させる監視プログラム。
【請求項７】
ＣＰＵモードをリアルモードに切替えるためのアクセスの発行を検出する監視ステップと、
ＯＳのブート後に検出した前記アクセスに対して、アクセス動作を無効にする無効ステップとを備えた情報処理監視方法。
【請求項８】
ＣＰＵモードをリアルモードに切替えるためのアクセスの発行を検出する監視手段、
ＯＳのブート後に検出した前記アクセスに対して、アクセス動作を無効にする無効処理手段、として情報処理装置を機能させるプログラムを記録したコンピュータ読み取り可能な記録媒体。
【請求項９】
ＯＳがブートされる主メモリと、主メモリ上にブートされたＯＳを実行するＣＰＵと、請求項１～４のいずれかに記載の情報処理監視装置とを備えた情報処理装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、コンピュータプログラムによって実行される情報処理動作を監視するセキュリティ技術に関する。
【背景技術】
【０００２】
昨今、不正かつ有害な動作を行う意図で作成された悪意のあるプログラム（以下、不当プログラム）によるリソース側へのアクセス、例えばファイルの改竄やシステムの設定変更等によってコンピュータのセキュリティが大きく損われている。今日、不当プログラムのリソースへのアクセスを監視し、不当動作を規制する各種の対応ソフトウェアが提案されている。
【０００３】

特許文献１には、リソースに対する処理要求であるＩ／Ｏプロセスを監視するステップと、アプリケーションから発行されたリソースに対する処理要求を一時保留するステップと、アプリケーションが正規のアプリケーションであるか否かをハッシュ値で認証するステップと、起動されたアプリケーションの認証結果が成功の場合にのみ、前記一時保留した処理要求による処理を許可するステップとを備えて、リソースの情報漏洩防止を図るアプリケーションの監視方法が記載されている。

【０００４】

特許文献２には、コンピュータ上で実行されるアプリケーションがハードディスクＨＤＤ等の記憶装置に記憶された情報にアクセスする時点で、オペレーティングシステム（ＯＳ）のフック機能を利用してこのアプリケーションをフックし、予め設定されたアクセス許容条件表の内容と照合することで判定し、マッチングしない場合にはウイルス等の不正なアクセスであるとして、記憶装置に記憶された情報のアプリケーションへの受け渡しを禁止するコンピュータの情報漏洩防止システムが記載されている。

【先行技術文献】

【特許文献】

【０００５】

【特許文献１】特開２００３－１０８２５３号公報

【特許文献２】特開２００７－１４０７９８号公報

【発明の開示】

【発明が解決しようとする課題】

【０００６】

特許文献１に記載されたアプリケーションの監視方法は、リソースに対する処理要求のプロセス毎に、当該アプリケーションのハッシュ値を算出し、算出したハッシュ値と予め記憶しているハッシュ値とを照合するものである。このように、リソースに対する処理要求時にハッシュ値による認証照合を試みるだけでは信頼性の高い監視に限界がある。

【０００７】

特許文献２に記載された情報漏洩防止システムは、ファイルシステムの入出力に関する機能（ＡＰＩ（ApplicationProgramming Interface）、シスコール（Syscall））を監視対象とするもので、アクセス許容条件として、コンピュータのユーザインタフェースを介して入力される、ファイルデータの書き込みなどの操作を表すＩＯタイプ、アプリケーション名、データファイル名、実行プログラムの実行制約条件（日時、範囲等）などの所定のチェック項目を、ホワイトリストであるアクセス許容条件表の内容と照合するものであるため、ホワイトリストとの照合処理のみでは、同様に監視に限界がある。

【０００８】

また、特許文献１，２は、リソースに対するＩＯアクセスの正当性を照合するべく監視対象としている点で充分とはいえない。さらに、特許文献１，２は、不当プログラムによるリソースへのアクセスを通して行われるファイルの改竄や削除、漏洩等を抑止しようとするものであり、ＣＰＵの動作状態に関するモードに起因して行われる虞のある不当な情報処理を抑止する技術については何等記載していない。

【０００９】

本発明は、上記に鑑みてなされたもので、ＣＰＵのモードの切替に起因する不当な情報処理を未然に抑止することが可能な情報処理監視装置、情報処理監視方法、監視プログラム、記録媒体及び情報処理装置を提供するものである。

【課題を解決するための手段】

【００１０】

本発明に係る情報処理監視装置は、ＣＰＵモードをリアルモードに切替えるためのアクセスの発行を検出する監視手段と、前記ＯＳのブート後に検出した前記アクセスに対して、アクセス動作を無効にする無効処理手段とを備えたものである。

【００１１】

また、本発明に係る監視プログラムは、ＣＰＵモードをリアルモードに切替えるための

アクセスの発行を検出する監視手段、ＯＳのブート後に検出した前記アクセスに対して、アクセス動作を無効にする無効処理手段、として情報処理装置を機能させるものである。
【００１２】
また、本発明に係る情報処理監視方法は、ＣＰＵモードをリアルモードに切替えるためのアクセスの発行を検出する監視ステップと、ＯＳのブート後に検出した前記アクセスに対して、アクセス動作を無効にする無効ステップとを備えたものである。
【００１３】
また、本発明に係る記録媒体は、ＣＰＵモードをリアルモードに切替えるためのアクセスの発行を検出する監視手段、ＯＳのブート後に検出した前記アクセスに対して、アクセス動作を無効にする無効処理手段、として情報処理装置を機能させるプログラムを記録したコンピュータ読み取り可能な記録媒体である。
【００１４】
また、本発明に係る情報処理装置は、ＯＳがブートされる主メモリと、主メモリ上にブートされたＯＳを実行するＣＰＵと、請求項１～４のいずれかに記載の情報処理監視装置とを備えたものである。
【００１５】
ＣＰＵは、ＯＳをアーキテクチャーとしてリアルモードで起動し、次いでプロテクトモードを経て、さらにより上位のビットモードに遷移して起動を終了する。起動後のＯＳは必要に応じてＣＰＵモードを切り換えて情報処理を実行する。ＣＰＵモードが特定のモードに切り替えられて、例えばリアルモードで不当な実行コードが発行され、ＯＳに識別されることなくレジスタの内容、環境設定値が不当に書き換えられる虞がある。そこで、ＣＰＵモードをリアルモードに切替えるためのアクセスである実行コードの発行を、ＯＳのブート後に検出したときは、前記アクセスは不当行為につながるものとしてに、アクセス動作を無効にするようにしている。これによって、ＣＰＵモードのリアルモードへの切替えを無効にすることで、リアルモードを利用することによる不当な情報処理を未然に抑止することができる。
【００１６】
また、監視モジュールをＯＳよりも高い特権レベルで主メモリにロードする監視モジュールロード手段を備え、前記監視モジュールは、前記監視手段及び前記無効処理手段を含むことを特徴とする。この構成によれば、ＯＳが発行するＣＰＵモードをリアルモードに切替えるためのアクセスを確実に検出することが可能となる。
【００１７】
また、前記アクセスは、ＣＰＵモードをプロテクトモードからリアルモードへ切替える実行コードである。この構成によれば、不当動作につながる動作を未然に防止することが可能となる。
【００１８】
また、前記監視手段は、ＣＰＵモードが６４ビットモードから３２ビットのプロテクトモードモードに切替えられた状態で、前記プロテクトモードから前記リアルモードへ切替える実行コードの発行を検出することを特徴とする。この構成によれば、ＣＰＵモードが６４ビットモードから３２ビットのプロテクトモードモードに切替えられた、次のアクセスであるプロテクトモードから前記リアルモードへの切替えを検出するので、より素早く不当動作を未然防止することが可能となる。
【００１９】
また、前記アクセスは、ページテーブルを無効にする実行コードである。この構成によれば、正当な動作としてはあり得ない、例外的な処理であって、不当動作の温床的なアクセスとなり得ることから、かかる実行コードを無効とすることで効果的な不当動作防止が図れる。
【発明の効果】
【００２０】
本発明によれば、ＣＰＵモードのリアルモードへの切替えを無効にすることで、リアル

モードを利用することによる不当な情報処理を未然に抑止することができる。
【図面の簡単な説明】
【００２１】
【図１】本発明に係る情報処理装置の一実施形態を示す概要構成図である。
【図２】チップセット及び主メモリであるＲＡＭのメモリマップを示す図である。
【図３】起動処理の手順を示すフローチャートである。
【図４】実行コードの実行手順を説明する図である。
【図５】監視モジュールによる監視動作処理（ＯＳブート後）の手順を示すフローチャートである。
【発明を実施するための形態】
【００２２】
図１は、本発明に係る情報処理装置の一実施形態を示す概要構成図である。本発明が適用される情報処理装置１００としては、汎用パーソナルコンピュータ、コンピュータを内蔵するサーバ装置、携行用の情報処理端末等、更にはネットワークを介して種々の目的で情報通信を行う機能を備えた各種の情報処理装置を含む。
【００２３】
情報処理装置１００は、マイクロコンピュタで構成され、プロセッサとしてのＣＰＵ１０及びチップセット１を備える。ＣＰＵ１０は、バスＢＡを介して、プログラムであるＢＩＯＳ（Basic Input/Output System）を格納するフラッシュＲＯＭ２１、補助記憶装置としてのハードディスクドライブ（ＨＤＤ：Hard Disc Drive）２２、主メモリとしてのＲＡＭ（Random Access Memory）３、入力部４１及び出力部４２と接続されている。
【００２４】
ＨＤＤ２２は、各種プログラム、ファイル及び必要なデータ類を格納する。ＨＤＤ２２は、複数のパーティションに分割されており、通常、例えばｗｉｎｄｏｗｓ（登録商標）のようなオペレーティングシステム（ＯＳ）のプログラム、及び当該ＯＳ下で動作する各種アプリケーションプログラム（ＡＰＰ）は、場所（パス）としてＣドライブ領域に格納されている。また、通常、各種ＡＰＰで利用（アクセス）可能なデータファイル等は、パスとしてＤドライブ領域に格納される。
【００２５】
ＲＡＭ３は、情報処理装置１００が起動する際に、フラッシュＲＯＭ２１に格納されているＢＩＯＳやＨＤＤ２２に格納されているプログラム等がロードされると共に、処理途中の情報が一時的に格納される。情報処理装置１００は、ＨＤＤ２２に格納されているプログラムファイル、データファイル等がＲＡＭ３にロードされ、ＣＰＵ１０によって実行されることで、文字通りの情報処理の他、必要に応じて、文書及び図形の作成や編集機能、情報通信、ブラウザ機能、電子決済、その他の種々の処理機能を実行する。
【００２６】
入力部４１は、テンキー等を備えたキーボードやマウス、又はタッチパネル等を含み、所要の情報の入力や処理の指示を行うものである。出力部４２は、画像を表示する表示部が想定される。なお、出力部４２としては、プリンタ部やインターネット等のネットワークに接続して情報の授受を行う通信部等でもよく、更にこれらが併設されたものでもよい。
【００２７】
図２は、ＲＡＭ３のメモリマップ及びチップセット１を示している。図２は、フラッシュＲＯＭ２１に格納されているＢＩＯＳ２１１、ＨＤＤ２２に格納されているＯＳ５、ＯＳ５の環境下で動作する各種のアプリケーションプログラムＡＰＰ１，ＡＰＰ２，…、ＨＤＤ２２のファイルシステム（ＮＴＦＳやＦＡＴ）外に格納されている動作監視部６としての監視モジュールがＲＡＭ３に読み出された状態の一例を示している。アプリケーションプログラムＡＰＰ１，ＡＰＰ２，…は、特権レベル（リングＲ＝３）のＡＰＰ領域３１に展開される。なお、ＡＰＩ（Application Programming Interface）もＡＰＰ領域３１内の所定領域に展開される。なお、ＡＰＩとは、複数のプログラムが共通に利用できる種

々の機能（ソフトウェア）の呼び出し時の手順やデータ形式などを定めた仕様をいう。

【００２８】

ＯＳ５は、特権レベル（リングＲ＝２）のカーネル領域３２に展開される。システムテーブル５１は、ＯＳを動作させるものである。システムテーブル５１は、ＣＰＵモードに対応して準備されるもので、ブート後の状態で、例えば６４ビットモードでは、ＯＳ５を動作させるための、対応するＧＤＴ（Global Descriptor Table）５１１、ＩＤＴ（InterruptDescriptor Table）５１２、ＴＳＳ（Task State Segment）５１３、及びＧＤＴ５１１からのリニアアドレスを物理アドレスに変換するＰＴＥ（Page Table Entry）５１４等を含む。動作監視部６は、特権レベル（リングＲ＝０）の監視処理領域３３に展開される。なお、図２では示していないが、動作監視部６を動作させるためのシステムテーブルが監視処理領域３３に展開される。

【００２９】

なお、ＯＳ５は、リングＲ＝１として領域３２に展開されてもよい。動作監視部６の特権レベルをＯＳ５に対して小さいリング値に設定することで、ＯＳ５の上位で、すなわち監視の下で動作させることが可能となる。

【００３０】

チップセット１は、ＣＰＵ１０の動作及び制御に必要とされる各種の環境設定値等が設定されるレジスタ群１２を含む。レジスタ群１２は、現在のＣＰＵ１０の状態を表すフラグレジスタやＭＳＲ（Model Specific Register）、データ格納用の汎用レジスタの他、メモリのアドレス指定等に関係するインデックスレジスタや特殊レジスタ、またメモリ管理方式に関係するセグメントレジスタCS等を含む。レジスタ群１２には、さらにコントロールレジスタ（CR0,CR4,CR3）、デバッグレジスタ（DR0～DR7）、ＭＳＲ（IA32E_EFERなど）、及びGDTR、IDTR、LDTRに格納されるＣＰＵ１０の環境設定値、ＭＳＲの環境設定値を含むことができる。また、他の環境設定値として、ＯＳがリアルモードの時、ソフトウェア割込みを使用してBIOS Functionを使用するBIOSジャンプ命令の環境設定値がある。このBIOS Functionの改ざんや不正使用の防止のため、後述するように起動後におけるリアルモードの割込みをフックする。

【００３１】

なお、ＲＡＭ３への各プログラムファイルの展開は、例えば以下のように、起動に際して行われる。図３を参照して情報処理装置１００の起動処理について説明する。まず、電源が投入されると、フラッシュＲＯＭ２１から、１６ビットのリアルモードのＢＩＯＳ２１１がＲＡＭ３の０～１MBiteの領域３４にロードされ、起動される（ステップＳ１）。次いで、利用可能な周辺機器の初期化を行うＰＯＳＴ（Power On Self Test）処理が実行される（ステップＳ３）。

【００３２】

続いて、動作監視部６である監視モジュールの領域３３へのロード処理が実行される（ステップＳ５）。まず、ＢＩＯＳ２１１によってＨＤＤ２２の先頭セクタからＭＢＲ（Master Boot Record）がＲＡＭ３にロードされ、次いでＣＰＵ１０の制御がＭＢＲに渡されて、ＭＢＲによって、ＨＤＤ２２のアクティブなパーティションテーブルに予め格納されている、監視モジュールロード手段である専用ローダ６０がＲＡＭ３にロードされる。この専用ローダによって一時的なＧＤＴが領域３４に作成され、このＧＤＴによって領域３３に対して特権レベルがリングＲ＝０に設定され、この状態で、ＨＤＤ２２のファイルスシステム外から、まず１６ビットのリアルモードから起動し、公知のようにアンリアルモードを利用して動作監視部６の監視モジュールが領域３３にロードされる。その後、専用ローダ６０は、３２ビットのプロテクトモードに遷移し、監視モジュールを実行する。監視モジュールは、さらにIA32e６４ビットモードに遷移してからＯＳを起動する。（なお、監視モジュールは、３２ビット状態でも所定の操作で動作可能）

次いで、動作監視部６の監視モジュールにより、監視環境の設定処理、すなわちシステムテーブル、割込ハンドラの作成等（共に図略）が行われる。なお、ＣＰＵモードとは、ＣＰＵ１０が扱うビット幅が１６ビット，３２ビット，６４ビットである各モードをいう

。
【００３３】
そして、ＣＰＵ１０の各モードでの動作監視部６のロード処理が終了すると（ステップＳ７）、ＣＰＵ１０の制御が動作監視部６に渡され、以降の監視動作及びＯＳ５のブート処理に移行する（ステップＳ９）。
【００３４】
まず、動作監視部６内に設定されたブートローダ（ＯＳローダ）が起動され（ステップＳ１１）、このブートローダによって領域３２へのＯＳ５のブート処理が行われる（ステップＳ１３）。
【００３５】
ＯＳ５のブート処理では、領域３２に特権レベルとしてリングＲ＝３の設定を行って、まず１６ビットのリアルモードでのＯＳ５が領域３２にロードされ、続いて特権レベルとしてリングＲ＝２を領域３２に設定して、３２ビットのプロテクトモードのＯＳ５、さらにIA32e６４ビットモードのＯＳ５がロードされる。また、各ＣＰＵモードでのシステムテーブル（図２では最後のシステムテーブル５１が示されている。）が作成される。このようにしてIA32e６４ビットモードのＯＳのブートが終了したと判断されると（ステップＳ１５）、その後、公知のように、ＯＳ５内のＡＰローダによって、必要なアプリケーションプログラムＡＰＰ１，ＡＰＰ２，…が領域３１に特権レベルとしてリングＲ＝３でロードされる（ステップＳ１７）。
【００３６】
また、動作監視部６の監視モジュールは、ＯＳ５のブート終了を受けて、例えばＯＳブート開始後の最初のページフォルトをトリガとして、ＣＰＵ１０のモードの切替えに対する監視（ＯＳブート後）を開始する。動作監視部６の監視モジュールは、ＯＳのブート終了までは、ＣＰＵ１０のモードが一時的にリアルモードに切替えられることを許可する一方、ＯＳ５のブート終了後は、図５で説明するように、ＣＰＵ１０のモードがリアルモードに切替えられることを、一般保護例外（＃ＧＰ）等を利用してフックすることで無効にしている。
【００３７】
図２に戻って、ＣＰＵ１０は、動作監視部６である監視モジュールがＨＤＤ２２から読み出されて、実行されることで、プロセス監視部６１、結果処理部６２及び規約情報部６３として機能する。なお、以降では、プロセス監視部６１の動作説明は、ＯＳブート後の監視に関するものとする。
【００３８】
プロセス監視部６１は、ＯＳ５によってＨＤＤ２２からＲＡＭ３に読み出された実行予定のアプリケーションプログラムＡＰＰ（例えばＡＰＰ１）に対して、後述するように監視を行う。より具体的には、プロセス監視部６１は、チップセット１のレジスタ群１２の特定のレジスタの環境設定値を書替える特定の実行コード（ＡＰＩ及びシスコール含む）の発行を少なくとも監視する。例えば、ＣＰＵ１０のモードを制御するレジスタに対して、ＣＰＵモードをリアルモードに切替えるための環境設定値を設定する実行コードを監視対象として含む。かかる環境設定値の書替えについては後述する。監視対象の実行コードは、例えば規約情報部６３に格納されている。
【００３９】
結果処理部６２は、監視対象の実行コードが実行されようとする時、当該実行コードの実行を無効（動作停止処理、禁止処理また正当内容への書替処理等）にする処理を行う。
【００４０】
実行コードの実行手順を説明する図４に示すように、例えばユーザによって実行が指示された実行プログラムの実行が開始されると、実行プログラムを構成している各実行コードが順次実行される。実行コードは、オペコードとそのパラメータとから構成されており、順次、ＣＰＵ１０のマイクロコード１１に命令セットとしてセットされる。動作監視部６は、監視対象の実行コードについては、当該実行コードがマイクロコード１１にセット

された後、実行直前に、動作を中断し、かつＮＴＤＬＬ．ＤＬＬによって、セットされた内容（実行コード、パラメータ）を取り込み、結果処理部６２によって当該実行コードの実行を無効（動作停止処理、禁止処理また正当内容への書替処理等）にする処理を行う。なお、本実施形態では、ＯＳ５はリングＲ＝２であるので、ＡＰＩ，Syscallなどの環境設定値への書替えアクセス（実行コード）は特権違反となり、フック対象とされる。

【００４１】

次に、図５の監視モジュールによる監視動作（ＯＳブート後）について説明する。ここでは、監視対象のアクセスは、ＣＰＵ１０のモードをリアルモードに切替えるための実行コードであり、かかる実行コードを無効にするものである。

【００４２】

ＣＰＵ１０は、起動（電源オンやリブート）時に、１６ビットのリアルモードでＢＩＯＳ２１１がロードされ、起動処理を実行する。ＯＳ５も同様に、ブートローダによって、ブート時には１６ビットのリアルモードからブートが開始され、３２ビットプロテクトモード、IA32e６４ビットモードの順で切り替わってブートが行われ、逆の場合には、ブート時の逆の手順を踏むことで、IA32e６４ビットモード、３２ビットプロテクトモード、１６ビットリアルモードの順で切り替わるようになっている。

【００４３】

通常、ＣＰＵ１０は、IA32e６４ビットモードか、３２ビットのプロテクトモードで動作しており、かかる６４ビットと３２ビットの間のモード切替は、容易に行われるようにされている。例えば、IA32e６４ビットモードから３２ビットプロテクトモードへの切替えはセグメントレジスタCSを３２ビットコードセグメントに切替えて、３２ビットプログラムに制御を移すことで可能となる。また、所定の手順を踏むことで、３２ビットプロテクトモードと１６ビットリアルモードの間の切替えも可能である。なお、ＯＳ５は、１６ビットプログラムがリングＲ＝０で動作するため、ＣＰＵ１０のモードが１６ビットリアルモードの状態での動作、１６ビットリアルモードへの遷移過程、IA32e６４ビットへの戻り遷移過程の動作を認識できない。しかも、ＲＡＭ３の領域３４である１６ビットリアルモード領域は、元々ＢＩＯＳ２１１等の格納領域であり、通常使用することはないことから、ＯＳ５のブート後は、リアルモードに移行することは通常では行わない。

【００４４】

一方、領域３４でのプログラムの挙動は、ＯＳ５で認識されないことから、不当な処理（データの削除、改竄、漏洩など）、乃至は不当なプログラムファイルを仕込む（例えば、ＢＩＯＳに不当なプログラムを感染させる）ことが可能となる。従って、通常のＣＰＵモードであるIA32e６４ビットモードで、バグ等のセキュリティホールを利用する仕掛けを種々の箇所に設けて長期間待機し、その後にＣＰＵ１０の制御を奪い取り、リアルモードへ移行しようとする悪意の試みがある。この種の待機型の悪意あるプログラム乃至はその実行の多くは、他のアンチウイルスソフトウエアで検知され、また無効化可能である。一方、ＣＰＵ１０のモードの切替えによる１６ビットリアルモードへの移行は、待機型とは異なり、所定の処理を経ることで可能となる。

【００４５】

図５において、監視モジュールは、まず、ＯＳ５側のＣＰＵ１０のモードが、IA32e６４ビットモードから３２ビットプロテクトモードへ変更されたか否かの判断を行う（ステップＳ２１）。ＣＰＵ１０のモードが、IA32e６４ビットモードのままであれば、変更なしとして、同様の判断処理が繰り返される。一方、ＣＰＵ１０のモードがIA32e６４ビットモードから３２ビットプロテクトモードへ変更されたのであれば、監視モジュールは、監視モジュール側のＣＰＵ１０のモードを、同様に、IA32e６４ビットモードから３２ビットプロテクトモードへ変更する（ステップＳ２３）。

【００４６】

次いで、監視モジュールは、ＯＳ５側のＣＰＵモードを１６ビットリアルモードに切替える実行コード（アクセス）の有無を判断する（ステップＳ２５）。具体的には、３２ビットプロテクトモードにおいて、マイクロコード１１で一時中断される順次の実行コード

が、ＣＰＵ１０を制御するためのコントロールレジスタCR0のページングフラグPGをオフにするアクセスの実行コードか否かの判断が行われ、そうであれば、当該アクセスを無効にする処理が実行される（ステップＳ２７）。一方、そうでない場合には、ステップＳ２７がスキップされ、すなわち当該実行コードはそのまま実行が許可される。

【００４７】

次いで、ＯＳ５側のＣＰＵ１０のモードが３２ビットプロテクトモードからIA32e６４ビットモードに戻ったか否かが判断され（ステップＳ２９）、戻ったのであれば、監視モジュール側のＣＰＵ１０のモードを３２ビットプロテクトモードからIA32e６４ビットモードに戻してステップＳ２１にリターンする。一方、ＣＰＵモードが戻っていなければ、ステップＳ２５に戻って、ＣＰＵモードを１６ビットリアルモードに切替える実行コード（アクセス）に対する監視と対応処理が繰り返される（ステップＳ２５～ステップＳ２９）。

【００４８】

なお、前記実施形態では、リアルモードへの切替のためのアクセスを無効とする方法として、コントロールレジスタCR0のページングフラグをオフ“０”にする段階のアクセスを監視したが、これに限定されず、かかる監視に続けて手順手続ルール（規約情報部６３に登録）として、ページテーブルを３２ビットプロテクトモード用に変換するアクセスとで、更にはIA32e６４ビットモードイネーブルフラグをオフ“０”にするアクセスを含めた態様としてもよい。このようにして、IA32e６４ビットモードから３２ビットプロテクトモードへの遷移の間にあるIA32eコンパチ３２ビットモードへの遷移処理を超えて、通常ではあり得ないような３２ビットプロテクトモードへの移行を監視することが可能となる。さらに、ＣＰＵモードをリアルモードに切替えるためのアクセスとして、３２ビットプロテクトモードから１６ビットリアルモードへ遷移するアクセスも含めて手順手続ルールで監視してもよい。例えば、３２ビットプロテクトモード用のIDTとGDTとをロードするアクセスを監視対象に含め、さらにコードセグメント（CS）をプロテクトモード用に移行するアクセス、TSSをプロテクトモードに移行するアクセス、コントロールレジスタCR0のページングフラグPG及びプロテクトモードフラグPEをオフ“０”にするアクセスをさらに監視対象に含めることができる。

【００４９】

なお、前記実施形態は、仮想化支援技術（ＶＴ－ｘ）未使用環境の態様で説明したが、これに限定されず、仮想化支援技術（ＶＴ－ｘ）使用環境の態様への適用も可能である。この場合、動作監視部６の監視モジュールをリングＲ＝－１、ＯＳ５をリングＲ＝０、ＡＰＰをリングＲ＝３と設定すればよい。この場合、コントロールレジスタCR0の変更のためのアクセスである実行コードは、規約情報部６３にフック対象として登録しておけばよい。実行コードは、例外、違反あるいは割込等の発生をトリガーとするいわゆるフック処理、またＡＰＩ及びシスコールをトリガーに含むことができる。

【００５０】

より具体的には、仮想化支援技術（ＶＴ－ｘ）使用環境では、前記監視対象の実行コードをフォールトイベントとして登録してフック対象とし、ＣＰＵ１０の制御をＶＭＭ（ホストソフトウェア）に渡してＶＭＸｒｏｏｔモードとするＥｘｉｔ命令と、逆にＣＰＵ１０の制御をゲストＯＳ側に渡してＶＭＸｎｏｎ－ｒｏｏｔモードとするＥｎｔｒｙ命令とを利用して監視と対応処理とを行うことが可能となる。それぞれの態様において、かかるアクセスを監視することが可能となる。なお、前記実施形態では、ＣＰＵ１０のモードを、ＯＳ５側をIA32e６４ビットモードから３２ビットプロテクトモードへ変更するのに対応させて、監視モジュール側も同様にＣＰＵモード変更処理を行ったが、仮想化支援技術（ＶＴ－ｘ）使用環境では、監視モジュール側はIA32e６４ビットモードのままでよい。

【符号の説明】

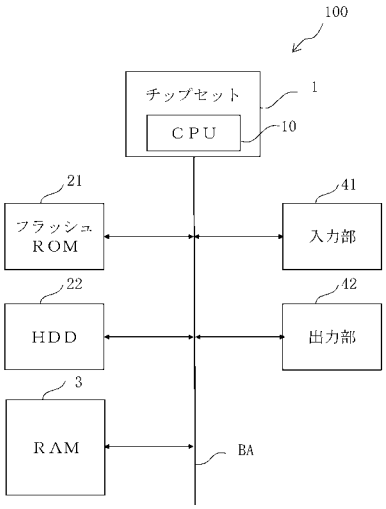
【００５１】

１００　情報処理装置

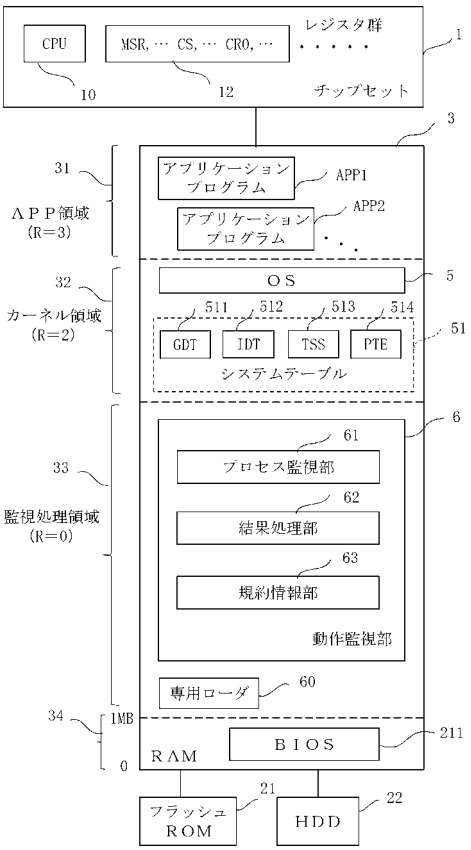
１　チップセット

１０　ＣＰＵ
１２　レジスタ群
２２　ＨＤＤ
３　ＲＡＭ
５　ＯＳ
６０　専用ローダ（監視モジュールロード手段）
６　動作監視部（監視モジュール）
６１　プロセス監視部（監視手段）
６２　結果処理部（無効処理手段）
６３　規約情報部（監視手段）

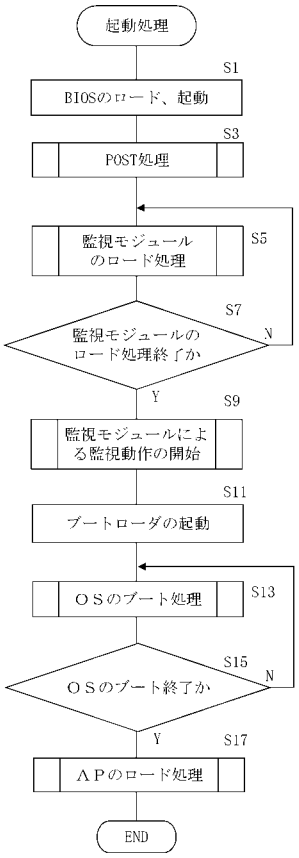
【図１】



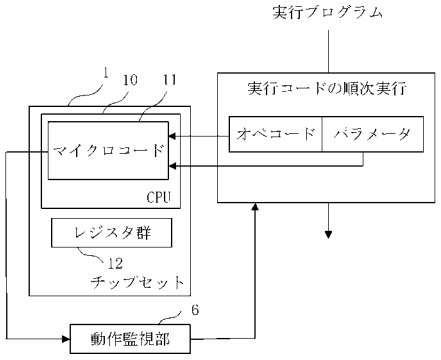
【図２】



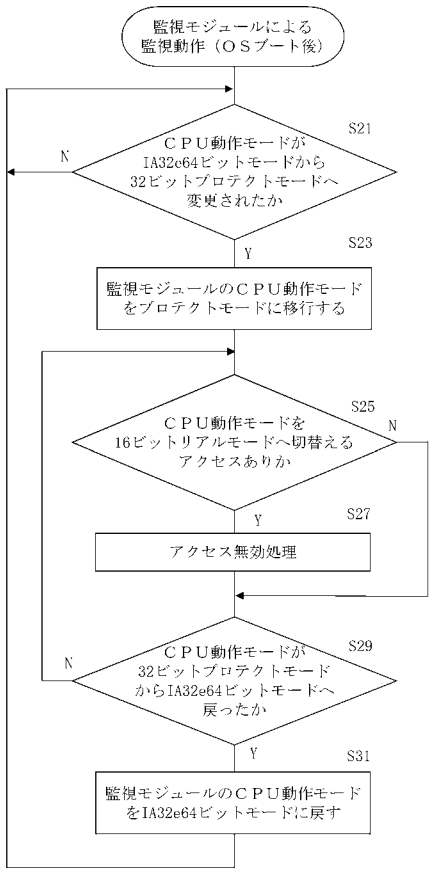
【図３】


起動処理
S1
BIOSのロード、起動
S3
POST処理
S5
監視モジュール
のロード処理
S7
監視モジュールの
ロード処理終了か
N
Y
S9
監視モジュールによ
る監視動作の開始
S11
ブートローダの起動
S13
ＯＳのブート処理
S15
ＯＳのブート終了か
N
Y
S17
ＡＰのロード処理
END

【図４】


実行プログラム
1
10
11
実行コードの順次実行
オペコード
パラメータ
マイクロコード
CPU
レジスタ群
12
チップセット
6
動作監視部

【図５】


監視モジュールによる
監視動作（ＯＳブート後）
S21
ＣＰＵ動作モードが
IA32e64ビットモードから
32ビットプロテクトモードへ
変更されたか
N
Y
S23
監視モジュールのＣＰＵ動作モード
をプロテクトモードに移行する
S25
ＣＰＵ動作モードを
16ビットリアルモードへ切替える
アクセスありか
N
Y
S27
アクセス無効処理
S29
ＣＰＵ動作モードが
32ビットプロテクトモード
からIA32e64ビットモードへ
戻ったか
N
Y
S31
監視モジュールのＣＰＵ動作モード
をIA32e64ビットモードに戻す