

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5058450号  
(P5058450)

(45) 発行日 平成24年10月24日(2012.10.24)

(24) 登録日 平成24年8月10日(2012.8.10)

(51) Int.Cl. F I  
G06F 11/00 (2006.01) G06F 9/06 630B

請求項の数 12 (全 18 頁)

(21) 出願番号	特願2005-128431 (P2005-128431)	(73) 特許権者	500046438 マイクロソフト コーポレーション アメリカ合衆国 ワシントン州 9805 2-6399 レッドモンド ワン マイ クロソフト ウェイ
(22) 出願日	平成17年4月26日(2005.4.26)	(74) 代理人	100077481 弁理士 谷 義一
(65) 公開番号	特開2005-327275 (P2005-327275A)	(74) 代理人	100088915 弁理士 阿部 和夫
(43) 公開日	平成17年11月24日(2005.11.24)	(72) 発明者	アンソニー ブラムフィールド アメリカ合衆国 98052 ワシントン 州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーシ ョン内
審査請求日	平成20年4月28日(2008.4.28)		
(31) 優先権主張番号	60/570, 124		
(32) 優先日	平成16年5月11日(2004.5.11)		
(33) 優先権主張国	米国 (US)		
(31) 優先権主張番号	10/881, 810		
(32) 優先日	平成16年6月30日(2004.6.30)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 効率的なパッチ当て

(57) 【特許請求の範囲】

【請求項 1】

ソフトウェアを修正するためにコンピューティングシステムにおいて実施される方法であって、

ソフトウェアの修正対象となった対象コンピュータシステムがパッチ配布サーバからパッチを受信するステップであって、前記パッチは修正モジュールに加えて、(a) 修正する実行可能モジュール、(b) 前記実行可能モジュールのパラメータ、(c) 前記パラメータに適用するテスト、(d) 前記パラメータが前記テストを満足しないときに実施されるテスト結果ハンドリングを指定するデータを含む、ステップと、

前記対象コンピュータシステムが、前記受信したパッチに基づき前記指定された実行可能モジュールを識別するステップと、

前記対象コンピュータシステムが、前記識別した実行可能モジュールに前記パッチを適用するとき、前記受信したパッチに基づき前記指定されたテストを前記識別された実行可能モジュールの前記指定されたパラメータに適用するステップと、

前記対象コンピュータシステムが、前記指定されたテストを前記指定されたパラメータに適用して前記指定されたテストが満足されない場合、前記指定されたテスト結果ハンドリングを実施するステップと、

前記指定されたテストが満足された、および、前記テスト結果ハンドリングが実施された場合、前記対象コンピュータシステムが、前記修正モジュールを前記指定された実行可能モジュールに適用するステップと

10

20

を備えたことを特徴とする方法。

【請求項 2】

前記テスト結果ハンドリングを実施するステップは、前記実行可能モジュールの実行を終了させることを特徴とする請求項 1 に記載の方法。

【請求項 3】

前記テスト結果ハンドリングを実施するステップは、前記指定されたパラメータを変更することを特徴とする請求項 2 に記載の方法。

【請求項 4】

前記対象コンピュータシステムが複数のパッチを受信することを特徴とする請求項 2 に記載の方法。

【請求項 5】

前記対象コンピュータシステムが受信した全てのパッチを保持することを特徴とする請求項 4 に記載の方法。

【請求項 6】

ユーザ許可無しに、前記実行可能モジュールを修正する修正モジュールを前記指定された実行可能モジュールに適用することを特徴とする請求項 1 に記載の方法。

【請求項 7】

前記パッチはパッチ構成コマンドをさらに含み、

前記対象コンピュータシステムが、前記パッチ構成コマンドに基づき前記パッチの動作を制御するステップであって、前記パッチ構成コマンドにおいて使用可能とされたときのみ前記指定された実行可能モジュールに前記修正モジュールを適用可能にする、ステップをさらに備えたことを特徴とする請求項 1 に記載の方法。

【請求項 8】

前記パッチ構成コマンドはグループポリシーに従って構成されることを特徴とする請求項 7 に記載の方法。

【請求項 9】

前記受信したパッチは署名されており、

前記パッチの署名を承認するステップであって、前記パッチの署名が承認された場合のみ前記修正モジュールを適用可能にする、ステップをさらに備えたことを特徴とする請求項 1 に記載の方法。

【請求項 10】

前記指定された実行可能モジュールはローダを用いて読み込み可能であり、

前記対象コンピュータシステムが前記指定された実行可能モジュールの読み込みに応答して前記指定された実行可能モジュールに前記修正モジュールを適用する前に呼び出された前記指定されたテストを実施するように前記ローダを用いることを特徴とする請求項 9 に記載の方法。

【請求項 11】

ソフトウェアを修正するためのコンピューティングシステムあって、

ソフトウェアを保持する記憶装置と、

前記コンピューティングシステムにおいてパッチ配布サーバからパッチを受信するパッチ受信部であって、前記パッチは修正モジュールに加えて、( a ) 修正する実行可能モジュール、( b ) 前記実行可能モジュールのパラメータ、( c ) 前記パラメータに適用するテスト、( d ) 前記パラメータが前記テストを満足しないときに実施されるテスト結果ハンドリングを指定するデータを含む、パッチ受信部と、

パッチ当てエージェントであって、

前記受信したパッチに基づき前記指定された実行可能モジュールを識別し、

前記識別した実行可能モジュールに前記パッチを適用するとき、前記受信したパッチに基づき前記指定されたテストを前記識別された実行可能モジュールの前記指定されたパラメータに適用し、

前記指定されたテストを前記指定されたパラメータに適用して前記指定されたテスト

10

20

30

40

50

が満足されない場合、前記指定されたテスト結果ハンドリングを実施し、

前記指定されたテストが満足される、および、前記テスト結果ハンドリングが実施された場合、前記修正モジュールを前記指定された実行可能モジュールに適用する、パッチ当てエージェントと  
を備えたことを特徴とするコンピュータシステム。

【請求項12】

コンピュータシステムにソフトウェアを修正させる方法を実施させるための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って、前記方法は、

ソフトウェアの修正対象となった対象コンピュータシステムがパッチ配布サーバからパッチを受信するステップであって、前記パッチは修正モジュールに加えて、(a)修正する実行可能モジュール、(b)前記実行可能モジュールのパラメータ、(c)前記パラメータに適用するテスト、(d)前記パラメータが前記テストを満足しないときに実施されるテスト結果ハンドリングを指定するデータを含む、ステップと、

前記対象コンピュータシステムが、前記受信したパッチに基づき前記指定された実行可能モジュールを識別するステップと、

前記対象コンピュータシステムが、前記識別した実行可能モジュールに前記パッチを適用するとき、前記受信したパッチに基づき前記指定されたテストを前記識別された実行可能モジュールの前記指定されたパラメータに適用するステップと、

前記対象コンピュータシステムが、前記指定されたテストを前記指定されたパラメータに適用して前記指定されたテストが満足されない場合、前記指定されたテスト結果ハンドリングを実施するステップと、

前記指定されたテストが満足される、および、前記テスト結果ハンドリングが実施された場合、前記対象コンピュータシステムが、前記修正モジュールを前記指定された実行可能モジュールに適用するステップと

を含むことを特徴とするコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、インストールされたコンピュータプログラムの動作の更新の分野に関する。

【背景技術】

【0002】

パッチ当てとは、アプリケーションプログラム、ユーティリティプログラム、オペレーティングシステムおよびオペレーティングシステムコンポーネント、装置ドライバなどを含む既にインストールされているプログラムを変更する処理である。パッチ当ては、プログラミングエラーの訂正、セキュリティリスクの減少または除去、または変更されたプログラムによって使用される論理の改善を含む様々な目的のためのプログラムの変更により有用であり得る。パッチ当ては、通常、パッチを当てられるプログラムを最初に供給した会社または他の組織によって開始される。

【0003】

インストールされたプログラムは、主に、実行可能コードモジュールからなる。一例として、米国ワシントン州レッドモンドのMicrosoft社のWINDOWS(登録商標)XPオペレーティングシステムで実行されるように設計された多くのプログラムが、主に「DLL」と称する実行可能コードモジュールからなる。パッチ当ての1つのよく知られている従来の手法は、パッチをあてられるインストール済みのプログラムを構成する実行可能コードモジュールの中から、パッチによって変更することが望まれるプログラムコードを含む実行可能コードモジュールを識別し; 所望の変更が行われた、識別された実行可能コードモジュールの新しいバージョンを作成し; 識別された実行可能コードモジュールの新しいバージョンを、インストーラプログラムと共に、パッチの適用を望む可能性があるユーザに配布することである。各ユーザは、パッチの適用を望むかどうかを判定し、適用を望む場合、インストーラプログラムを実行する。これによって、識別された実行可能

10

20

30

40

50

コードモジュールのオリジナルバージョンが、識別された実行コードモジュールの新しいバージョンに置換される。

【0004】

【特許文献1】米国特許出願公開第10/307,902号明細書

【発明の開示】

【発明が解決しようとする課題】

【0005】

パッチ当ての従来手法は、複数の重要な不利益を有する。多くの場合、パッチの受け取りおよび適用に関連する負担を増すという不利益をもたらす。ある場合には、負担の増加が、一部のユーザによる一部のパッチの適用を遅らせ、一部のユーザによる一部のパッチの適用を妨げもする。そのようなパッチ適用の遅延および阻害は、ある場合には、特にセキュリティリスクを減少または除去するために設計されたパッチの場合には、ユーザにとって深刻で否定的な結果をもたらす可能性がある。

10

【0006】

パッチ当てに関する従来手法の1つの不利益は、単一のプログラムに対する単一の変更をもたらすために複数のパッチを作成し、配布しなければならない一般的な状況に関連する。ある場合、パッチをあてられるプログラムは、プログラムがそこで実行されるように設計されたオペレーティングシステムまたはオペレーティングシステムバージョンごとにおよび/またはプログラムの自然言語バージョンごとに異なる「フレイバ(flavor)」など、特定の実行可能コードモジュールの複数の異なるフレイバを有する。識別された実行可能コードモジュールが、そのような実行可能コードモジュールである場合、前述したパッチ作成配布プロセスを、識別された実行可能コードモジュールのフレイバごとに繰り返さなければならない。ユーザは、識別された実行可能コードモジュールの適当なフレイバ用のパッチを選択し、適用しなければならない。結果として生じた多数のパッチをソートし、各ユーザのコンピュータシステムのアプリケーションに正しいパッチ群を選択することは、非常に負担になる可能性がある。この状態は、時折「パッチ地獄」と呼ばれる。ある場合には、管理者は、各対象システムにインストールされた実行可能モジュールバージョン群を識別するインベントリデータベースを維持しなければならない。これは、各対象システムに適当な従来パッチを選択するのに使用される。

20

【0007】

パッチ当てに対する従来手法のもう1つの不利益は、配布されるパッチのサイズが大きいことに関する。実行可能コードモジュールが、メガバイト単位のサイズを有することがめずらしくない。その故、単一のパッチが匹敵するサイズを有し、配布と保管が手に負えなくなり、一部のユーザにとって配布および保管が不可能になる。この問題は、複数のフレイバを有するパッチのために、複合的になる可能性がある。さらに、各従来パッチに、通常は、置換実行可能モジュール全体が含まれるので、従来パッチの適用は、コードチャーン(code churn)の問題に寄与する可能性がある。

30

【0008】

パッチ当てに対する従来手法のさらにもう1つの不利益は、一部のユーザが、実動コンピュータシステムに適用する前にパッチをテストする必要があることに関する。ある場合、コンピュータシステムへのパッチのインストールは、パッチに含まれる識別された実行可能コードモジュールの新しいバージョンが新しいプログラミングエラーを導入する場合、またはそれが適用されるコンピュータシステムで動作する別のプログラムとの新しい予期されない相互作用を引き起こす場合など、悪い結果を有する可能性がある。したがって、多くの場合、そのデータおよび動作の維持が重要である実動システムにパッチを適用する前に、ユーザは、まず、テストシステムにパッチを適用して、そのパッチを実動システムに適用しても安全であるか否かを評価する。パッチのそのような別々のテストは、パッチ当てに関連する負担を増やす。さらに、パッチが適用されてから十分に後に、従来パッチが(アプリケーション互換性の問題または新しいエクスポイト脆弱性など)問題を起こす場合に、その問題をパッチまでさかのぼって追跡することが困難になる可能性がある。

40

50

## 【 0 0 0 9 】

パッチ当てに対する従来の手法のさらなる不利益は、パッチに含まれるインストーラの動作に関する。多くの場合、実行プログラムの一部である実行可能コードモジュールを置換するために、インストーラは、まずそのプログラムの実行を終わらせなければならない。また、ある場合、そのような置換は、コンピュータシステムを再起動しなければ完了することができない。この両ステップは、パッチをあてられるコンピュータシステムの使用にかなりの混乱を引き起こす可能性がある。

## 【 0 0 1 0 】

パッチ当てに対する従来の手法のもう1つの不利益は、「ホットフィックス」とも称する「プライベートフィックス (private fix)」が、その実行可能モジュールの顧客の正しいサブセットに以前に発行された実行可能モジュールにパッチを当てることに関する。その場合、ユーザがそのホットフィックスを適用したか否かによって各ユーザに依存する実行可能コードモジュールを、異なる新しいバージョンに置換する従来のパッチの配布で出会う問題のために、それに替えて、ユーザがホットフィックスを適用したか否かに無関係に実行可能モジュールの単一の新しいバージョンに置換する単純な従来のパッチの配布が通常である。その新しいバージョンがホットフィックスを実装する場合、そのパッチは、ホットフィックスを受け取ることを意図されていない顧客にホットフィックスを課す。その一方で、新しいバージョンがホットフィックスを実施しない場合、ホットフィックスを受け取ることが意図された顧客にホットフィックスを与えない。

## 【 0 0 1 1 】

パッチ当てに対する従来の手法のもう1つの不利益は、特定のダイナミックリンクライブラリなど、特定の実行可能モジュールに頼るソフトウェア製品のインストーラが、多くの場合、対象コンピュータシステムのファイルシステムにおいて標準的でない位置にその実行可能モジュールを保管することにより、その実行可能モジュールを「隠す」という事実に関する。したがって、特定の対象システムに、パッチをあてるべき実行可能モジュールのコピーが含まれるか否かを判定し、そうである場合にそれが対象コンピュータシステムのファイルシステムの何処にあるかを判定することが、しばしば困難または不可能である。また、いくつかのソフトウェア製品は、そのインストーラによってインストールされた実行可能モジュールバージョンの「カタログ」を維持する。ソフトウェア製品は、特定の実行可能モジュールのバージョンのカタログ内の表示の正しさに頼る可能性がある。そのような信用は、従来のパッチが、カタログを更新せずに、カタログで識別される実行可能モジュールのバージョンをその実行可能モジュールの新しいバージョンに置換する場合に失われる。

## 【 0 0 1 2 】

パッチ当てに対する従来の手法のもう1つの不利益は、パッチを当てられる実行可能モジュールが対象コンピュータシステムにインストールされる前に、そのパッチを適用することが不可能であるという事実から生じる。その結果、パッチをあてられる実行可能モジュールが、その実行可能モジュールの従来のパッチが受け取られた後に対象コンピュータシステムにインストールされる場合、そのパッチが実行可能モジュールに適用される可能性は低い。

## 【 0 0 1 3 】

パッチ当てに対する従来の手法のもう1つの不利益は、そのようなパッチが、通常、対象コンピュータシステムにログインしたユーザによって、自由な変更の許可を有する管理アカウントを使用してのみ適用できることである。この目的で管理アカウントにログインすることは、対象コンピュータシステムの諸態様を変更することを試み、それを行う自由な許可を必要とする対象コンピュータシステムに存在するウイルスに対して、対象コンピュータシステムを脆弱にする可能性がある。

## 【 0 0 1 4 】

パッチ当てに対する従来の手法のもう1つの不利益は、従来のパッチが、使用不能にすることが困難または不可能である可能性があり、実行可能モジュールの置換を逆転するか

10

20

30

40

50

、またはシステムレジストリに対する1つまたは複数の変更を逆転することを必要とする  
ことである。

【0015】

したがって、上で述べたパッチ当てに対する従来の手法の不利益の一部またはすべてを  
克服する、パッチ当てに対する新しい手法が、高い有用性を有するはずである。

【課題を解決するための手段】

【0016】

インストールされたコンピュータプログラムコードにパッチをあてるソフトウェア機構  
(「本機構」)を提供する。ある態様において、本機構は、インストールされた機能にパ  
ラメータテストおよびテスト結果ハンドリングを追加する。他の態様では、本機構は、あ  
る場合に、インストールされた機能の実行の流れの任意の位置で、インストールされた機  
能に様々な他の種類の機能性を追加する。

10

【0017】

ある態様において、パッチごとに、本機構は、パッチをあてられる各コンピュータシ  
ステム(すなわち、各「対象コンピュータシステム」)に、テストを実行すべき点、実行さ  
れるテストの識別、および1つまたは複数の異なるテスト結果に応答して行動する方法の  
指定を配布する。ある態様において、本機構は、その使用をパッチで指定できるパラメ  
ータ検証および他のテストの標準的な集合を提供する。例えば、パッチは、特定の機能につ  
いて、その機能の特定のパラメータにある値が含まれない場合、その実質的な実行が開始  
される前に、その機能の呼出しが失敗すべきであることを指定することができる。もう1  
つのパッチが、特定の機能について、特定のパラメータが指定された最大の長さを超える  
長さを有する場合、その機能の実行が進行を許可される前に、そのパラメータを指定され  
た最大の長さまでに切り詰めるべきであることを指定することができる。多くのセキュリ  
ティエクスプロイトは、一機能のコードのオリジナルバージョンでは阻まれなかったが、  
その機能が危険な状態を作り、またはこれを利用させるパラメータ値を用いて機能と呼び  
出させることを当てにする。多くの場合、そのようなエクスプロイトは、そのようなパッ  
チを使用して、機能がそのようなパラメータ値を用いて実行されないようにすることによ  
り防ぐことができる。ある態様において、パッチは、ファイルから読み取られる値または  
ユーザによって入力される値など、機能パラメータ以外の値のテストを指定する。

20

【0018】

ある態様では、自動化されたパッチ当てエージェントが、自動的に各パッチを受け取り  
、検証し、可能なアプリケーションに関するパッチテーブルに保管する。ある態様におい  
て、各パッチは、パッチが受け取られた時に対象コンピュータシステムに既に読み込まれ  
ている、パッチを当てられる実行可能モジュールのすべてのインスタンスに適用される。  
この手法を、本明細書では「ホットパッチ当て」と称し、これによって、パッチが受け取  
られた直後にパッチを有効にすることが可能になり、パッチを当てられる実行可能モジ  
ュールがディスクに保管されているか否かを本機構が判定できる必要はない。ある態様にお  
いて、各受け取られたパッチは、パッチを当てられる実行可能モジュールのディスクイメ  
ージに適用され、その結果、ディスクイメージが将来読み込まれる時に、読み込まれるデ  
ィスクイメージにパッチが含まれるようになる。この手法を、本明細書では「コールドパ  
ッチ当て」と称し、これによって、パッチが、複数のセッションにまたがって持続できる  
ようになる。ある態様において、本機構は、ホットパッチ当てとコールドパッチ当ての両  
方を実行する。ある態様において、各パッチは、パッチを当てられる実行可能モジ  
ュールがオペレーティングシステムのロードによって読み込まれるたびに、パッチを当てられる  
実行可能モジュールに適用される。この手法を、本明細書では「読み込み時パッチ当て」  
と称する。ある態様において、各パッチは、パッチを当てられる機能が呼び出されるた  
びに、パッチをあてられる実行可能モジュールに適用される。この手法を、本明細書では「  
呼出しインターセプトパッチ当て」と称する。読み込み時パッチ当ておよび呼出しインター  
セプトパッチ当ての両方が、(1)パッチを当てられる実行可能モジュールがディスク  
のどこに保管されているかを本機構が判定できることを必要とせず、(2)特定のパッチ

30

40

50

の即座の可逆性を容易にし、かつ(3)実行可能モジュールのディスクイメージの変更を必要としない。

【0019】

ある態様において、本機構は、ユーザまたは管理者が適用されるパッチの動作を構成できるようにする。例として、そのような構成に、特定の適用されるパッチに関して：パッチによって指定されたテストを、実行がパッチについて指定された点に達したときに実行するか否か；パッチによって指定されたテスト結果ハンドリングを実行するか、または無視するか；ならびに／あるいはテストの実行および／またはその結果をログ記録し、警告メッセージで表示するか否かなどを含めることができる。これらの態様で、本機構は、まずロギングを使用可能にし、結果ハンドリングを使用不能にすることによって、実動コンピュータシステムでパッチをテストできるようにする。これらの態様で、本機構は、さらに、その結果ハンドリングを使用可能にした後にパッチの動作を「冗長モード」でログ記録できるようにして、パッチが、アプリケーション互換性問題または他のIT問題などの問題を引き起こしている実例を識別するのを支援する。これらの態様は、パッチが問題を起こしていることが発見された場合、適用された後にパッチを素早く使用不能にすることも可能にする。本機構のいくつかの態様は、受け取られ、対象コンピュータシステムに保管されたパッチ群から単純にパッチを削除することによって、パッチを素早く使用不能にすることも可能にする。

10

【0020】

ある態様において、本機構は、「データドリブン」パッチ当て手法を使用する。この手法では、パッチに、コードではなく、小さい人間読取り可能なテキストまたはXML文書などのデータが含まれる。このデータは、テストを実行すべき点、実行すべきテストの識別、および1つまたは複数のテスト結果に回答して行動する方法を指定する。そのような態様において、パッチ当てエージェントが、データドリブンパッチを受け取り、パッチによって指定されたテストおよびテストハンドリングを追加する。ある態様において、本機構は、「コードドリブン」パッチ当て手法を使用する。この手法では、各パッチに、パッチをあてられる実行可能モジュールに追加される短いプログラムが含まれる。このプログラム自体は、本機構の標準パラメータテスト機能呼び出すことによってテストを実行し、テストハンドリングを実行する。データドリブンパッチ当てまたはコードドリブンパッチ当てを使用することによって、時折、パッチをあてられる実行可能モジュールのすべてのフレイバに単一のパッチで対処することが可能になる。

20

30

【0021】

ある態様において、本機構は、各パッチに署名して、(1)パッチが承認されたソースからであること、および(2)パッチが承認されたソースによって作成された時以降、パッチの内容が変更されていないことの両方を実証する。

【0022】

ある態様において、本機構は、すべてのパッチをすべての対象コンピュータシステムに配布し、対象コンピュータシステムのパッチ当てエージェントが、その対象コンピュータシステムでどのパッチを適用すべきか、および適用する方法を、対象コンピュータシステムの特성에基づいて自動的に判定する。これによって、ユーザおよび管理者が、従来、パッチの選択および適用に関連した多くの負担、および正確な現在のインベントリデータベースの維持の負担から解放される。例えば、この特性に、パッチをあてられる実行可能モジュールのどのバージョンが対象コンピュータシステムにインストールされているかを含めることができる。これらの態様において、本機構は、特定の実行可能モジュールのホットフィックスされたフレイバおよびホットフィックスされていないフレイバの異なる処理を指定するパッチを配布し、特定の実行可能モジュールのホットフィックスを犠牲にすることも、実行可能モジュールにパッチをあてる時にホットフィックスを遍在させることも必要としなくすることによって、通常、ホットフィックスによって引き起こされる類の問題を克服することができる。

40

【0023】

50

ある態様において、パッチ当てエージェントは、特定のパッチによってパッチを当てられる実行可能モジュールが、そのパッチを受け取った時に対象システムにインストールされているか否かに無関係に、受け取られたすべてのパッチを対象システムに保管する。本機構は、多くの場合、パッチを当てられる実行可能モジュールの読み込み、またはパッチを当てられる機能の呼出しにตอบสนองしてパッチを適用する。そのため、本機構は、パッチが対象システムで受け取られた後に対象システムにインストールされた実行可能モジュールにパッチを適用することができる。また、パッチは、パッチを当てられる実行可能モジュールのアンインストールおよびその後の再インストールを超えて残存することができる。

【 0 0 2 4 】

ある態様において、パッチ当てエージェントは、オペレーティングシステムサービスで実施される。これらの態様で、本機構は、パッチを適用するのに必要なすべての許可をパッチ当てエージェントに与える。これらの態様は、ユーザが広範囲の変更許可を有する管理者アカウントを使用して対象コンピュータシステムにログインする必要を無くし、これによって、対象コンピュータシステムに存在するすべてのウィルスに、対象コンピュータシステムの重要な態様を変更するより大きな機会を与えない。そのため、従来のパッチが適用される時に通常は課せられるセキュリティリスクを減らす。

【 0 0 2 5 】

本機構によって使用されるパッチは、通常、比較的小さく、したがって、伝送および保管に関して適度なリソース要件を課す。また、本機構によって使用されるパッチは、パッチをあてられるソフトウェアの挙動を少数の明確に定義された形で変更するので、本機構は、コードチャーンの問題を減らすことに資する。

【 発明を実施するための最良の形態 】

【 0 0 2 6 】

図 1 に、本機構を実施できる適切なコンピューティングシステム環境 1 0 0 の例を示す。コンピューティングシステム環境 1 0 0 は、適切なコンピューティング環境の一例にすぎず、本機構の使用または機能性の範囲に関する制限を提案することを意図されたものではない。コンピューティング環境 1 0 0 を、例示的なオペレーティング環境 1 0 0 に示された構成要素のいずれかまたはその組合せに関する依存性または要件を有するものと解釈すべきでない。

【 0 0 2 7 】

本機構は、多数の他の汎用または特殊目的のコンピューティングシステム環境またはコンピューティングシステム構成と共に動作する。本機構と共に使用するのに適する可能性がある周知のコンピューティングシステム、コンピューティング環境、および/またはコンピューティング構成の例に、パーソナルコンピュータ、サーバコンピュータ、ハンドヘルド装置、ラップトップ装置、タブレット装置、マルチプロセッサシステム、マイクロプロセッサに基づくシステム、セットトップボックス、プログラム可能な民生用電子機器、ネットワーク PC、ミニコンピュータ、メインフレームコンピュータ、上記のシステムまたは装置のいずれかを含む分散コンピューティング環境、および類似物が含まれるが、これに限定されない。

【 0 0 2 8 】

本機構を、プログラムモジュールなど、コンピュータによって実行されるコンピュータ実行可能命令の全般的な文脈で説明することができる。一般に、プログラムモジュールには、特定のタスクを実行し、または特定の抽象データ型を実施する、ルーチン、プログラム、オブジェクト、コンポーネント、データ構造などが含まれる。本機構は、通信ネットワークを介してリンクされたりリモート処理装置によってタスクが実行される分散コンピューティング環境で実践することもできる。分散コンピューティング環境では、プログラムモジュールを、メモリ記憶装置を含むローカルおよび/またはリモートのコンピュータ記憶媒体に配置することができる。

【 0 0 2 9 】

図 1 を参照すると、本機構を実装する例示的なシステムに、コンピュータ 1 1 0 の形の

10

20

30

40

50



汎用コンピューティング装置が含まれる。コンピュータ110のコンポーネントに、処理装置120、システムメモリ130、およびシステムメモリを含む様々なシステムコンポーネントを処理装置120に結合するシステムバス121が含まれる。システムバス121は、メモリバスまたはメモリコントローラ、周辺バス、および様々なバスアーキテクチャのいずれかを使用するローカルバスを含む複数の種類のバスの構造のいずれかとすることができる。限定ではなく例として、そのようなアーキテクチャに、ISA (Industry Standard Architecture) バス、MCA (マイクロチャネルアーキテクチャ) バス、EISA (Enhanced ISA) バス、VESA (Video Electronics Standards Association) ローカルバス、およびメザンバスとも称するPCI (Peripheral Component Interconnect) バスが含まれる。

10

**【0030】**

コンピュータ110に、通常は、様々なコンピュータ読取り可能な媒体が含まれる。コンピュータ読取り可能な媒体は、コンピュータ110によってアクセスできるすべての使用可能な媒体とすることができ、揮発性および不揮発性、取外し可能および固定の両方を含む使用可能な媒体を含む。限定ではなく例として、コンピュータ読取り可能な媒体に、コンピュータ記憶媒体および通信媒体を含めることができる。コンピュータ記憶媒体に、コンピュータ読取り可能な命令、データ構造、プログラムモジュール、または他のデータなどの情報を保管する任意の方法またはテクノロジーで実施された、揮発性および不揮発性、取外し可能および固定の媒体が含まれる。コンピュータ記憶媒体に、RAM、ROM、EEPROM、フラッシュメモリ、または他のメモリテクノロジー、CD-ROM、デジタル多用途ディスク(DVD)、または他の光学ディスク記憶、磁気カセット、磁気テープ、磁気ディスク記憶、または他の磁気記憶装置、あるいは所望の情報の保管に使用でき、コンピュータ110によってアクセスできる他のすべての媒体が含まれるが、これに限定されない。通信媒体によって、通常は、搬送波または他のトランスポート機構などの変調されたデータ信号内のコンピュータ読取り可能な命令、データ構造、プログラムモジュール、または他のデータが実装され、通信媒体には、すべての情報配布媒体が含まれる。用語「変調されたデータ信号」は、信号内で情報をエンコードする形でその特性の1つまたは複数を設定または変更された信号を意味する。限定ではなく例として、通信媒体に、有線ネットワークまたは直接配線接続などの有線媒体と、音響、RF、赤外線、および他の無線媒体などの無線媒体が含まれる。上記のいずれの組合せも、コンピュータ読取り可能な媒体の範囲に含まれなければならない。

20

30

**【0031】**

システムメモリ130に、読取専用メモリ(ROM)131およびランダムアクセスメモリ(RAM)132などの揮発性メモリおよび/または不揮発性メモリの形のコンピュータ記憶媒体が含まれる。起動中などにコンピュータ110内の要素の間での情報の転送を助ける基本ルーチンを含む基本入出力システム133(BIOS)が、通常はROM131に保管される。RAM132には、通常は、処理装置120から即座にアクセス可能、かつ/または処理装置120によって現在操作中のデータおよび/またはプログラムモジュールが含まれる。限定ではなく例として、図1に、オペレーティングシステム134、アプリケーションプログラム135、他のプログラムモジュール136、およびプログラムデータ137を示す。

40

**【0032】**

コンピュータ110に、他の取外し可能/固定の、揮発性/不揮発性のコンピュータ記憶媒体も含めることができる。単に例として、図1に、固定の不揮発性磁気媒体から読み取られ、またはこれに書き込むハードディスクドライブ141、取外し可能不揮発性磁気ディスク152から読み取られ、またはこれに書き込む磁気ディスクドライブ151、CD-ROMまたは他の光媒体などの取外し可能不揮発性光ディスク156から読み取られ、またはこれに書き込む光ディスクドライブ155を示す。例示的なオペレーティング環境で使用できる他の取外し可能/固定の、揮発性/不揮発性のコンピュータ記憶媒体に、

50

磁気テープカセット、フラッシュメモリカード、デジタル多用途ディスク、デジタルビデオテープ、ソリッドステートRAM、ソリッドステートROM、および類似物が含まれるが、これに限定されない。ハードディスクドライブ141は、通常は、インターフェース140などの固定のメモリアンターフェースを介してシステムバス121に接続され、磁気ディスクドライブ151および光ディスクドライブ155は、通常、インターフェース150などの取外し可能メモリアンターフェースによってシステムバス121に接続される。

#### 【0033】

前述し、図1に示したドライブおよびそれに関連するコンピュータ記憶媒体は、コンピュータ110のコンピュータ読取り可能な命令、データ構造、プログラムモジュール、および他のデータの記憶を提供する。図1では、例えば、ハードディスクドライブ141が、オペレーティングシステム144、アプリケーションプログラム145、他のプログラムモジュール146、およびプログラムデータ147を保管するものとして図示されている。これらのコンポーネントを、オペレーティングシステム134、アプリケーションプログラム135、他のプログラムモジュール136、およびプログラムデータ137と同一のまたは異なるもののいずれとすることもできることに留意されたい。オペレーティングシステム144、アプリケーションプログラム145、他のプログラムモジュール146、およびプログラムデータ147は、少なくとも異なるコピーであることを示すために、異なる符号を与えられている。ユーザは、タブレットまたは電子デジタイザ164、マイクロホン163、キーボード162および、一般にマウス、トラックボール、またはタッチパッドと称するポインティングデバイス161などの入力装置を介してコンピュータ110にコマンドおよび情報を入力することができる。図1に示されていない他の入力装置に、ジョイスティック、ゲームパッド、衛星パラボラアンテナ、スキャナ、または類似物を含めることができる。上記のおよび他の入力装置は、しばしば、システムバスに結合されたユーザ入力インターフェース160を介して処理装置120に接続されるが、パラレルポート、ゲームポート、またはユニバーサルシリアルバス(USB)などの他のインターフェースおよびバス構造によって接続することができる。モニター191または他の種類のディスプレイ装置も、ビデオインターフェース190などのインターフェースを介してシステムバス121に接続される。モニター191を、タッチスクリーンパネルまたは類似物と一体化することもできる。モニターおよび/またはタッチスクリーンパネルを、タブレットタイプパーソナルコンピュータのように、コンピューティング装置110が組み込まれるハウジングに物理的に結合できることに留意されたい。さらに、コンピューティング装置110などのコンピュータに、スピーカ195およびプリンタ196など、出力周辺インターフェース194または類似物を介して接続できる他の周辺出力装置も含めることができる。

#### 【0034】

コンピュータ110は、リモートコンピュータ180などの1つまたは複数のリモートコンピュータへの論理接続を使用して、ネットワーク化された環境で動作することができる。リモートコンピュータ180は、パーソナルコンピュータ、サーバ、ルータ、ネットワークPC、ピア装置、または他の一般的なネットワークノードとすることができ、通常、上でコンピュータ110に関して説明した要素の多くまたはすべてが含まれるが、図1には、メモリ記憶装置181だけを示した。図示の論理接続に、ローカルエリアネットワーク(LAN)171および広域ネットワーク(WAN)173が含まれるが、他のネットワークも含めることができる。そのようなネットワーキング環境は、オフィス、会社規模のコンピュータネットワーク、イントラネット、およびインターネットで一般的なものである。例えば、本機構で、コンピュータシステム110に、そこからデータが移行されるソースマシンを含めることができ、リモートコンピュータ180に、宛先マシンを含めることができる。しかし、ソースマシンおよび宛先マシンを、ネットワークまたは他の手段によって接続する必要がなく、その代わりに、データを、ソースプラットフォームによる書き込みと宛先プラットフォームによる読み取りが可能な任意の媒体を介して移行でき

10

20

30

40

50

ることに留意されたい。

【0035】

L A Nネットワーク環境で使用される時に、コンピュータ110は、ネットワークインターフェースまたはネットワークアダプタ170を介してL A N 171に接続される。W A Nネットワーク環境で使用される時に、コンピュータ110に、通常は、インターネットなどのW A N 173を介する通信を確立する、モデム172または他の手段が含まれる。モデム172は、内蔵または外付けとすることができるが、ユーザ入力インターフェース160または他の適当な手段を介してシステムバス121に接続することができる。ネットワーク化された環境では、コンピュータ110に関して図示されたプログラムモジュールまたはその一部を、リモートメモリ記憶装置に保管することができる。限定ではなく例として、図1に、メモリ装置181に常駐するものとしてリモートアプリケーションプログラム185を示す。図示のネットワーク接続が例示的であり、コンピュータの間の通信リンクを確立する他の手段を使用できることを理解されたい。

10

【0036】

様々な機能性およびデータが、特定の形で配置された特定のコンピュータシステムにあるものとして図1に示されているが、当業者は、そのような機能性およびデータを、異なる配置のコンピュータシステムにまたがって様々な他の形で分散できることを理解するであろう。前述したように構成されたコンピュータシステムが、通常は本機構の動作をサポートするのに使用されるが、当業者は、本機構を、様々な種類および構成の、様々なコンポーネントを有する装置を使用して実施できることを理解するであろう。

20

【0037】

図2に、本機構によるコンピュータシステム間の通常の日データ交換を示すデータフローを示す。図2に示すコンピュータシステム(コンピュータシステム210、220、221、222、230、231、および232)は、通常、図1に示され、図1に関して述べたコンポーネントのいくつかまたはすべてを有する。パッチ配布サーバにおいて、本機構が、1つまたは複数のパッチを生成する。これらのパッチ201は、パッチ配布サーバから、管理サーバ220および230などの1つまたは複数の管理サーバに送信される。各管理サーバは、そのパッチを、対象コンピュータシステム221、222、231、および232などの1つまたは複数の対象コンピュータシステムに転送する。ある実施形態(図示せず)において、パッチ配布サーバは、1つまたは複数の対象コンピュータシステムに直接に、または単一の管理サーバを介するよりも間接的なルートを通じて、パッチを送信する。対象コンピュータシステムで受け取られたパッチは、下記で詳細に説明するように、対象コンピュータシステムで処理される。管理サーバは、1つまたは複数の対象コンピュータシステムにパッチ構成コマンド202を送信することができ、対象コンピュータシステムは、そのパッチ構成コマンドを適用して、特定のパッチの動作を再構成する。下記で詳細に述べるように、パッチは、完全に使用不能にすることができ、あるパッチが使用不能にされない場合、その動作の通知およびそのテスト結果ハンドリングのそれぞれを、独立に使用可能または使用不能にすることができる。その動作の通知が使用可能にされるとき、通知は、対象コンピュータシステムでローカルに表示もしくは保管することができ、または、通知203として適当な管理サーバに送信することができる。

30

40

【0038】

図3に、通常、新しいパッチを受け取り、および処理するために本機構によって実行されるステップの流れ図を示す。ステップ301で、本機構は、パッチを受け取る。ステップ301で受け取られたパッチは、データドリブンパッチまたはコードドリブンパッチのいずれかとするすることができる。データドリブンパッチのサンプルを、下の表1に示す。

【0039】

【表 1】

```

1   <Softpatch Patch="Q382429">
2   <AffectedApplication AffectedExe="sqlservr.exe">
3   <AffectedVersion Version="9.*">
4   <AffectedModules Name="SQLSORT.DLL">
5   <Version "8.0.*, 9.*">
6   <Function Name="SsrpEnumCore" Address="0x0802E76B"
7   Param="2" Paramtype="LPSTR">
8   <Filter MaxByteLength="60" />
9   <Resolution ActionType="BOOL" Action="FALSE" />
10  </Function>
11 </Version>
12 <Version "10.*, 11.*">
13 <Function Name="SsrpEnumCore" Address="0x0802D283"
14 Param="2" Paramtype="LPSTR">
15 <Filter MaxByteLength="128" />
16 <Resolution ActionType="BOOL" Action="FALSE" />
17 </Function>
18 </Version>
19 </AffectedModules>
20 </AffectedVersion>
21 </AffectedApplication>
22
23 <Signature Hash="MD5" Signature="C509-64AA-9161-8C52-
24 9F6D-BF5A-AEF2-ECE1-0038-34D1"/>
25 </Softpatch>

```

表 1

## 【 0 0 4 0 】

行 1 に、パッチの一意識別子が含まれる。行 2 によって、このパッチによって影響されるアプリケーションが識別される。行 3 によって、このパッチによって影響されるアプリケーションバージョンが識別される。行 4 によって、このパッチによって影響される実行可能モジュールが識別される。行 5 によって、パッチ当ての指示が提供される、影響される実行可能モジュールの 2 つのバージョン（バージョン 8 . 0 . \* および 9 . \* ）が識別される。行 6 ~ 10 に、実行可能モジュールのこの 2 つのバージョンに関するパッチ当て指示が含まれる。行 6 ~ 7 によって、パッチをあてられる機能、実行可能モジュール内のそのアドレス、パッチによってテストされるパラメータ、およびテストされるパラメータの型が識別される。行 8 によって、行 6 ~ 7 で識別されたパラメータに対し、その長さが 60 バイトを超えるか否かを判定するためにテストを行うべきであることが示される。行 9 によって、テストが成功の場合にこの機能の呼出しが失敗すべきであることが示される。行 12 によって、パッチ当て指示が提供される、影響される実行可能モジュールのもう 2 つのバージョン（バージョン 10 . \* および 11 . \* ）が識別される。行 13 ~ 17 に、実行可能モジュールのこの 2 つのバージョンに関するパッチ当て指示が含まれる。行 13 ~ 14 によって、パッチを当てられる機能、実行可能モジュール内のそのアドレス、パッチによってテストされるパラメータ、およびテストされるパラメータの型が識別される。バージョン 10 . \* および 11 . \* について行 13 ~ 14 で識別される実行可能モジュール内のパッチを当てられる機能のアドレスが、バージョン 8 . 0 . \* および 9 . \* について行 6 ~ 7 で識別されたパッチを当てられる機能のアドレスと異なることが分かる。行 15 によって、行 13 ~ 14 で識別されたパラメータをテストして、その長さが 128 バイトを超えないか否かを判定すべきであることが示される。行 16 によって、テストが成功する場合、この機能の呼出しが失敗すべきであることが示される。パッチは、パッチを当てられる機能の呼出しの失敗、例外の送出、パッチを当てられた実行可能モジュールが実行されているプロセスの終了、または誤った値の訂正（長すぎるストリングの切詰など）を含む様々な結果ハンドリングアクション型を指定することができる。行 23 ~ 25 に、パッチのソースを識別すると同時に、ソースから出た後にパッチが変更されていないことを検証するパッチの署名が含まれる。

## 【 0 0 4 1 】

下記の表 2 に、上記の表 1 に示すパッチのコードドリブンバージョンが含まれる。

## 【 0 0 4 2 】

## 【表 2】

1	00411A7E	push	3Ch
2	00411A80	mov	eax,dword ptr [str]
3	00411A83	push	eax
4	00411A84	call	ValidateStringLength (411082h)
5	00411A89	add	esp,8
6	00411A8C	movzx	ecx,al
7	00411A8F	test	ecx,ecx
8	00411A91	je	411A9Ah
9	00411A93	jmp	foo+2 (411AD2h)
10	00411A9A	xor	eax, eax
11	00411A9C	ret	

10

表 2

## 【 0 0 4 3 】

行 1 ~ 3 で、テスト機能のパラメータをスタックにプッシュする。行 4 で、テスト機能呼び出す。行 5 ~ 8 で、テスト機能の戻りコードに基づいて分岐する。テスト機能が成功の場合に、行 9 で、ジャンプして、パッチをあてられた機能の本体の実行を開始する。テスト機能が失敗する場合に、行 10 ~ 11 で、失敗の結果コードをスタックにプッシュし、パッチを当てられた機能からパッチを当てられた機能の呼出し元に戻る。読み易くするため、表 2 では、検証可能な署名、パッチ構成フラグの現在の値についてテストする命令、およびパッチを当てられた機能のコードの先頭から再配置された命令を含む、一部のコードドリブンパッチに存在する詳細が省略されている。

20

## 【 0 0 4 4 】

ある実施形態において、両方の型のパッチは、パッチを当てられる実行可能モジュールの 1 つまたは複数のバージョンごとに、実行可能モジュールの特定のインスタンスがそのバージョンの正しいコピーであることを検証するのに使用できるファイル署名を含む追加情報を含むことができる。そのようなファイル署名は、例えば、実行可能モジュールバージョン全体のサイズもしくはチェックサム、または、実行可能モジュールがパッチを当てられるオフセットなどの実行可能モジュール内の特定の点にあると期待されるコードとすることができる。

30

## 【 0 0 4 5 】

ステップ 302 で、パッチに有効な署名が署名されている場合、本機構は、ステップ 303 で継続する。そうでない場合、本機構は、ステップ 301 で継続して、次のパッチを受け取る。ステップ 303 で、本機構は、パッチをローカルパッチテーブルに追加する。ステップ 304 で、本機構は、デフォルト構成に送ることによるなどのように、パッチの初期構成を初期化する。

## 【 0 0 4 6 】

図 4 は、本機構によって使用されるものに典型的なサンプルパッチテーブルを示すデータ構造図である。パッチテーブル 400 に、行 401 および 402 などの行が含まれ、これらの行の各々は、パッチから抽出されたパッチ識別子を含むパッチ識別子列 411、名前など、パッチを当てられる実行可能モジュールを識別する情報を含む実行可能モジュール列 412、パッチが適用される、列 412 で識別された実行可能モジュールのバージョンのすべてを識別する実行可能モジュールバージョン列 413、パッチを当てられる機能が呼び出されるたびに、このパッチによって指定されるテストを実行されるべきか否かの現在の構成値を含むテスト実行可能列 414、このパッチのテストが実行されるたびに通知を生成されるべきか否かの現在の構成値を含むテスト実行通知可能列 415、このパッチのテストが成功するたびに通知を生成されるべきか否かの現在の構成値を含むテスト結果通知実行可能列 416、このパッチのテストが失敗した時にパッチの結果ハンドリングを実施されるべきか否かの現在の構成値を含むテスト結果ハンドリング実行可能列 417

40

50

、ならびにテストおよびテストが失敗するたびに実行されるテスト結果ハンドリングを指定する、パッチ自体へのポイントを含むパッチ列 4 1 8 に分割される。ある実施形態において、図に示すようにパッチへのポイントを含むのではなく、パッチ列 4 1 8 にパッチ自体が直接に含まれる。特定のパッチテーブルは、すべてのコードドリブンパッチ、すべてのデータドリブンパッチ、またはコードドリブンパッチとデータドリブンパッチの組合せなど、様々な型のパッチを含み、またはこれを指すことができる。

**【 0 0 4 7 】**

ステップ 3 0 5 で、本機構が、受け取ったパッチをパッチテーブルに追加し、その構成を初期化したならば、そのパッチは、本機構による実行可能モジュールへの自動的適用に使用可能である。ステップ 3 0 5 で、本機構は、参照によって組み込まれた本特許明細書に記載のもの、ならびにリアルタイム機能呼出しインターセプション、および/または ( 1 ) 既に読み込まれた実行可能モジュール、( 2 ) 実行可能モジュールの 1 つまたは複数のディスクイメージ、もしくは ( 3 ) オペレーティングシステムのロードによって読み込まれる実行可能モジュールのインスタンスのコード書き換えを含むパッチを適用する様々な手法を使用することができる。ステップ 3 0 5 の後に、本機構は、ステップ 3 0 1 において継続して、次のパッチを受け取る。

10

**【 0 0 4 8 】**

図 5 は、通常、特定のパッチの構成命令を更新するために本機構によって実行されるステップを示す流れ図である。ステップ 5 0 1 で、本機構は、管理者からなど、特定のパッチの構成命令を受け取る。ある実施形態において、そのような構成命令は、グループポリシーを使用して管理者が生成することができる。ステップ 5 0 2 で、本機構は、受け取った命令に従って、パッチテーブル内のパッチの構成を更新する。ステップ 5 0 2 の後に、本機構は、ステップ 5 0 1 で継続して、次の構成命令を受け取る。

20

**【 0 0 4 9 】**

図 6 は、通常、パッチによって指定されるパラメータ検証を実行するために本機構によって実行されるステップを示す流れ図である。ステップ 6 0 1 で、パッチを当てられた機能呼び出す。ステップ 6 0 2 で、呼び出された機能に影響するパッチについてテストが実行可能にされている場合、本機構は、ステップ 6 0 3 で継続し、そうでない場合には、本機構は、ステップ 6 0 1 で継続して、パッチを当てられた機能の次の呼出しを処理する。ステップ 6 0 3 で、テスト実行通知がこのパッチについて実行可能にされている場合に、本機構は、ステップ 6 0 4 で継続し、そうでない場合に、本機構は、ステップ 6 0 5 で継続する。ステップ 6 0 4 で、本機構は、テストが実行されたことの通知を生成する。ステップ 6 0 4、6 0 8、および 6 1 0 に、テストが満足されたことを表示する対象コンピュータシステムでの表示または保管、および/あるいはリモートコンピュータシステムでの表示またはログ記録に関するリモートコンピュータシステムへのそのような表示の送信を含むことができる。

30

**【 0 0 5 0 】**

ステップ 6 0 5 で、本機構は、パッチによって指定される検証テストを実行する。ある実施形態において、ステップ 6 0 5 に、本機構によってテストに使用される標準ルーチンのグループの 1 つを呼び出すことが含まれる。ステップ 6 0 6 で、ステップ 6 0 5 で実行されたテストが満足される場合、本機構は、ステップ 6 0 1 で継続する。そうでない場合、本機構は、ステップ 6 0 7 で継続する。ステップ 6 0 7 で、テスト結果通知がこのパッチについて実行可能にされている場合、本機構は、ステップ 6 0 8 で継続する。そうでない場合、本機構は、ステップ 6 0 9 で継続する。ステップ 6 0 8 において、本機構は、テストが満足されなかったことの通知を生成する。ステップ 6 0 9 で、このパッチについてテスト結果ハンドリングが実行可能にされている場合、本機構は、ステップ 6 1 0 で継続する。そうでない場合、本機構は、ステップ 6 0 1 で継続する。ステップ 6 1 0 において、本機構は、パッチによって指定されるテスト結果ハンドリングを実行する。ステップ 6 1 0 の後に、本機構は、ステップ 6 0 1 で継続する。

40

**【 0 0 5 1 】**

50

当業者は、上記で説明した機構を、様々な形で簡単に適合または拡張できることを理解されたい。例えば、本機構を使用して、様々な異なる種類のパッチを様々な形で様々な目的の様々な型の実行可能モジュールの様々な位置に適用することができる。また、本明細書では、失敗する時に問題を示す値検証テストを含むものとしてパッチを説明したが、本機構は、成功する時に問題を示す値検証テストを使用して実施することもできる。いくつかの実施形態において、各テストに、その成功または失敗が問題を示すか否かの表示が付随する。前述の説明では、好ましい実施形態を参照したが、本発明の範囲は、請求項およびこれに記載された要素によってのみ定義される。

【図面の簡単な説明】

【0052】

10

【図1】本機構を実施できる適切なコンピューティングシステム環境の例を示す図である。

【図2】本機構によるコンピュータシステム間のデータの通常の交換を示すデータフロー図である。

【図3】通常、新しいパッチを受け取り、処理するために本機構によって実行されるステップを示す流れ図である。

【図4】本機構によって使用されるものに典型的なサンプルパッチテーブルを示すデータ構造図である。

【図5】通常、特定のパッチの構成命令を更新するために本機構によって実行されるステップを示す流れ図である。

20

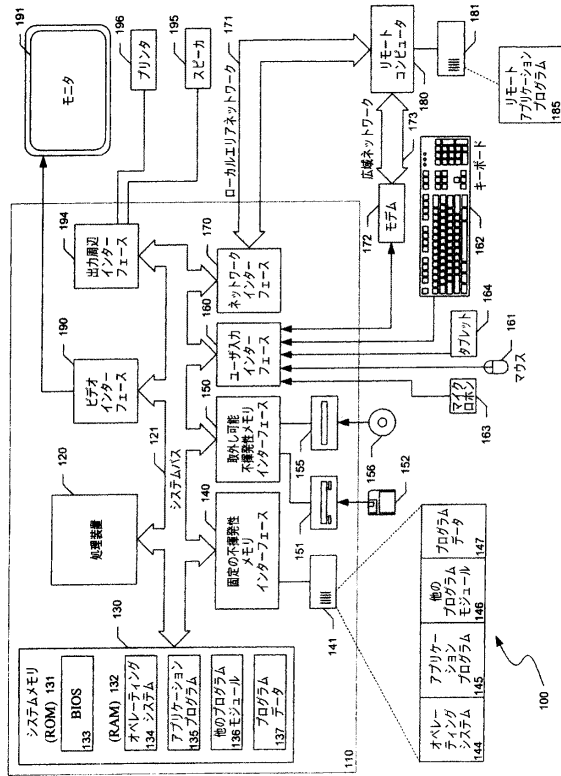
【図6】通常、パッチによって指定されるパラメータ検証を実行するために本機構によって実行されるステップを示す流れ図である。

【符号の説明】

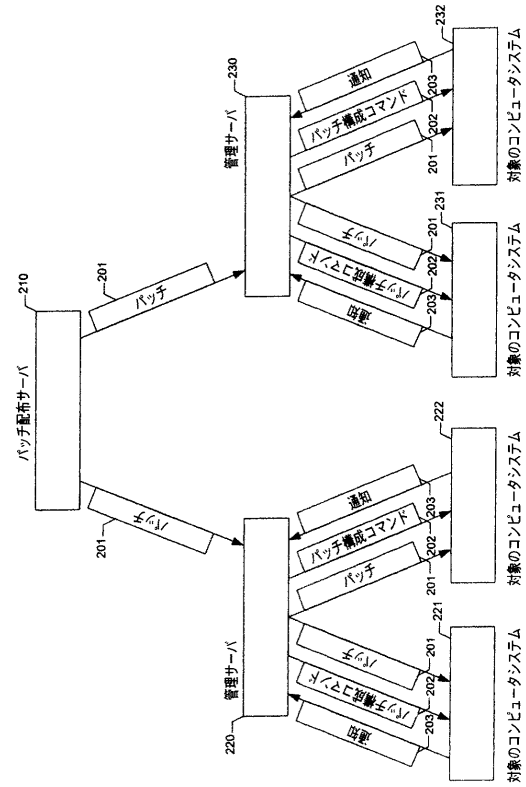
【0053】

- 141 ハードディスクドライブ
- 151 磁気ディスクドライブ
- 155 光ディスクドライブ
- 181 メモリ記憶装置

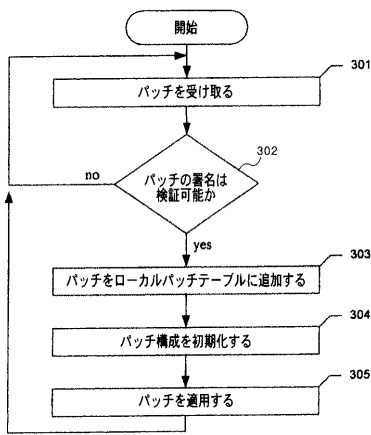
【図1】



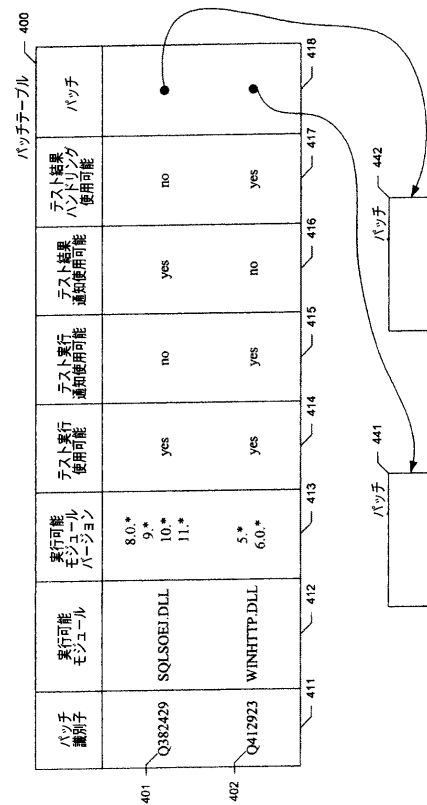
【図2】



【図3】

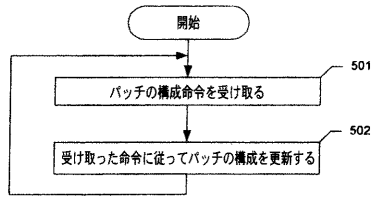


【図4】

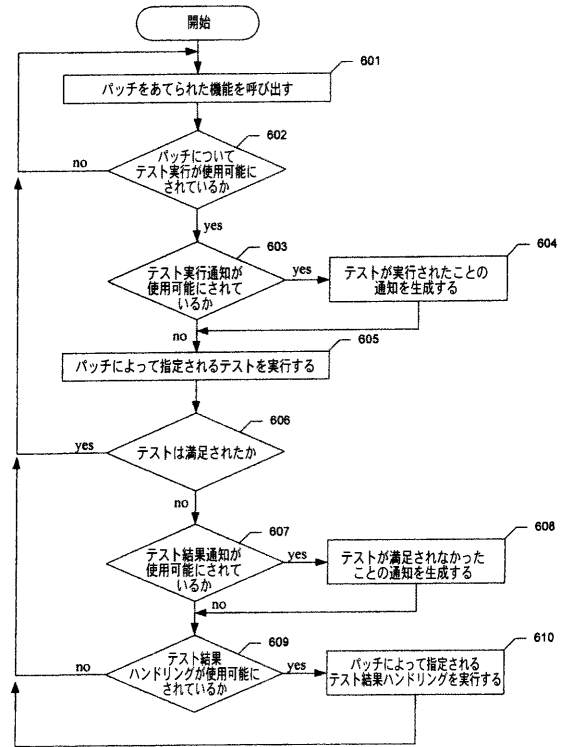




【図5】



【図6】



## フロントページの続き

- (72)発明者 ギラド ゴラン  
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ  
イクロソフト コーポレーション内
- (72)発明者 ジェーソン ガームズ  
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ  
イクロソフト コーポレーション内
- (72)発明者 ソード エー．アルシパニ  
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ  
イクロソフト コーポレーション内
- (72)発明者 スコット エー．フィールド  
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ  
イクロソフト コーポレーション内

審査官 久保 光宏

- (56)参考文献 特開平1-112427(JP,A)  
特開2003-44305(JP,A)

- (58)調査した分野(Int.Cl., DB名)  
G06F9/06,  
G06F11/00,  
G06F9/44,  
G06F9/445,  
CSDB(日本国特許庁)