(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4928721号
(P4928721)
(45)発行日　平成24年5月9日(2012.5.9)　(24)登録日　平成24年2月17日(2012.2.17)

(51)Int.Cl.　F I
G06F 12/10 (2006.01)　G06F 12/10 505Z
G06F 12/10 541

請求項の数 22　(全 23 頁)

(21)出願番号　特願2004-289030 (P2004-289030)
(22)出願日　平成16年9月30日 (2004.9.30)
(65)公開番号　特開2005-135396 (P2005-135396A)
(43)公開日　平成17年5月26日 (2005.5.26)
審査請求日　平成19年9月28日 (2007.9.28)
(31)優先権主張番号　10/697,197
(32)優先日　平成15年10月30日 (2003.10.30)
(33)優先権主張国　米国 (US)

(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　９８０５２－６３９９　レッドモンド　ワン　マイクロソフト　ウェイ
(74)代理人　100077481
弁理士　谷　義一
(74)代理人　100088915
弁理士　阿部　和夫
(72)発明者　アーネスト　エス．コーエン
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

最終頁に続く

(54)【発明の名称】アドレス変換制御のためのシャドウ・ページテーブル

(57)【特許請求の範囲】
【請求項１】
アドレス変換マップのためのシャドウページを作成する方法を実行するためのコンピュータ実行可能命令を格納したコンピュータ読取り可能な記録媒体であって、
前記アドレス変換マップは、ページディレクトリーおよび複数のページテーブルを含み、前記ページディレクトリーは、複数のページテーブルへのリンクを含み、前記ページテーブルの各々は、複数のデータページへのリンクを含み、前記ページディレクトリーおよび前記ページテーブルの各々は前記データページの１つに格納され、前記方法は、
前記複数のページテーブルの少なくとも１つに対し、前記複数のページテーブルの１つに基づいて第１のシャドウページテーブルを作成するステップであって、前記第１のシャドウページテーブルは、
前記第１のシャドウページテーブル中の少なくとも１つのエントリーが、前記複数のページテーブルの内の第１のページテーブルの中の、そのエントリーの対応するリンクと異なるデータページにリンクする点と、
前記第１のシャドウページテーブルは、前記複数のページテーブルの内の前記第１のページテーブルの中の対応するリンクが、読み出し／書き込みである１または複数の読み出し専用リンクを含む点と
の少なくとも１つにおいて、前記複数のページテーブルの内の前記第１のページテーブルと異なる、ステップと、
前記ページディレクトリーに基づいてシャドウページディレクトリーを作成するステッ

プであって、前記ページディレクトリーは、前記複数のページテーブルの１つへのリンクを含み、前記シャドウページディレクトリーは、前記複数のページテーブルの前記１つへのリンクの代わりに前記シャドウページテーブルへのリンクを含む、ステップとを備え、

前記複数のページテーブルの内の前記第１のページテーブルを使用するソフトウェアオブジェクトは、前記ソフトウェアオブジェクトが前記複数のページテーブルの内の前記第１のページテーブルに以前に書き込んだデータであって、前記複数のページテーブルのうちの前記第１のページテーブル内に存在するデータに依存する非アドレスマッピング動作を実行し、前記データは、前記複数のページテーブルの内の前記第１のページテーブル内に存在するが、前記第１のシャドウページテーブル内に存在しない特性を有し、前記第１のシャドウページテーブルは、前記ソフトウェアオブジェクトに関するアドレスを変換するために用いられ、前記複数のページテーブルの内の前記第１のページテーブルは、前記ソフトウェアオブジェクトに関するアドレスを変換するために用いられないことを特徴とするコンピュータ読取り可能な記録媒体。

【請求項２】

アクセスが許可されるページを定義するポリシーがメモリへのアクセスを管理し、仮想アドレスに適用された前記アドレス変換マップに基づく前記メモリへのアクセスは、前記ポリシーに基づいて許可されず、前記仮想アドレスに適用された前記シャドウページディレクトリーおよび前記第１のシャドウページテーブルは、前記ポリシーに基づいてアクセスが許可されることを特徴とする請求項１に記載のコンピュータ読取り可能な記録媒体。

【請求項３】

前記データページの各々は、メモリの特定のフレームに格納され、前記ページディレクトリーは第１のフレームに格納され、前記方法は、

前記第１のフレームと異なる第２のフレームにある前記ページディレクトリーのコピーを保持するステップと、

前記第１のフレームに前記シャドウページディレクトリーを格納するステップと

をさらに備えることを特徴とする請求項１に記載のコンピュータ読取り可能な記録媒体。

【請求項４】

前記ページディレクトリーは、第１のサイズのページへのリンクを備え、前記第１のサイズのページは、複数の第２のサイズのページを備え、前記方法は、

前記複数の第２のサイズのページへのリンクを含む第２のシャドウページテーブルを作成するステップであって、前記シャドウページディレクトリーは、前記第２のシャドウページテーブルへのリンクを含む、ステップをさらに備えることを特徴とする請求項１に記載のコンピュータ読取り可能な記録媒体。

【請求項５】

メモリの使用を管理するためのシステムであって、

個々にアドレス割り当て可能であって、読み出し・書き込み可能な複数のページを格納したメモリであって、前記複数のページの各々は、それに関係付けられる物理アドレスを有する、メモリと、

仮想アドレスと前記複数のページの前記物理アドレスとの間のマッピングを定義するアドレス変換データ構造であって、前記アドレス変換データ構造は、前記複数のページへのリンクを含む複数のページテーブル、および前記複数のページテーブルへのリンクを含む複数のページディレクトリーを含み、前記ページディレクトリーおよび各ページテーブルは、前記複数のページの１つに格納され、前記複数のページの各々は、それに関係付けられる物理位置記述子を有し、前記ページディレクトリーおよび前記ページテーブル中の前記リンクの各々は、前記物理位置記述子に基づきページの１つを特定する、アドレス変換データ構造と、

前記複数のページの内の第１のページにアクセスするためのリクエストを受けとるメモリマネジャーであって、前記リクエストは、前記仮想アドレスに基づいて前記複数のページの内の前記第１のページを特定する、メモリマネジャーとを備え、

前記メモリマネジャーは、前記アドレス変換データ構造のシャドウ表現を含むデータに基づいて、前記複数のページの内の前記第１のページの物理アドレスに前記仮想アドレスを変換し、前記シャドウ表現は、前記ページディレクトリーの少なくとも１つあるいは前記ページテーブルの１つの代替のバージョンを含み、前記代替のバージョンは、代替のバージョンが基づくページと異なる物理位置記述子を有するページに格納され、前記アドレス変換データ構造を使用するソフトウェアオブジェクトは、前記ソフトウェアオブジェクトの現在の動作に関して、前記ソフトウェアオブジェクトが前記アドレス変換データ構造に以前に書き込んだデータであって、前記複数のページのうちの前記第１のページ内に存在するデータに依存する非アドレスマッピング動作を実行し、前記データは、前記アドレス変換データ構造内に存在するが、前記アドレス変換データ構造の前記シャドウ表現内に存在しない特性を有し、前記アドレス変換データ構造の前記シャドウ表現は、前記ソフトウェアオブジェクトに関するアドレスを変換するために用いられ、前記アドレス変換データ構造は、前記ソフトウェアオブジェクトに関するアドレスを変換するために用いられないことを特徴とするシステム。

【請求項６】

前記アドレス変換データ構造の前記シャドウ表現は、少なくとも１つのリンクに関して前記アドレス変換データ構造とは異なることを特徴とする請求項５に記載のシステム。

【請求項７】

前記ページディレクトリーおよび前記ページテーブルに含まれるリンクの各々は１または複数の属性を含み、前記シャドウ表現の中の少なくとも１つのリンクは、少なくとも１つの属性に関する前記アドレス変換データ構造において対応するリンクとは異なることを特徴とする請求項６に記載のシステム。

【請求項８】

アクセスが許可されるページを定義するポリシーは、メモリのアクセシビリティを管理し、前記システムは、

前記アドレス変換データ構造に基づいて前記シャドウ表現を作成し、前記シャドウ表現が仮想アドレスに基づいてメモリにアクセスするために使用される場合、前記ポリシーに基づいてアクセスが許可されることを保証するメモリアクセス制御マネジャーをさらに備えることを特徴とする請求項５に記載のシステム。

【請求項９】

前記ポリシーが、アクセス不可として前記メモリの一部を定義し、前記メモリアクセス制御マネジャーは、前記シャドウ表現が前記メモリの一部に対する仮想アドレスを公開しないことを保証することを特徴とする請求項８に記載のシステム。

【請求項１０】

前記ポリシーは、読み出し可能であるが書き込み不可として前記メモリの一部を定義し、前記メモリアクセス制御マネジャーは、前記シャドウ表現が、読み出し専用として前記メモリの一部にマークする１または複数の属性を含むことを保証することを特徴とする請求項８に記載のシステム。

【請求項１１】

前記メモリアクセス制御マネジャーは、前記シャドウ表現が、読み出し専用として、（１）アドレス変換データ構造、および（２）前記シャドウ表現の少なくとも１つを格納するメモリの部分を読み出し専用としてマークする１または複数の属性を含むことを保証することを特徴とする請求項８に記載のシステム。

【請求項１２】

メモリへのアクセスリクエストを実行する方法であって、

メモリユニットに読み出しまたは書き込みをするためのリクエストを受けるステップであって、前記リクエストは、仮想アドレスに基づいて前記メモリユニットを特定する、ステップと、

仮想アドレスと物理アドレスとの間の関係を定義するマップの表現に基づいて前記メモリユニットにアクセスするステップであって、前記マップは、ページディレクトリーまた

は複数のテーブルを含み、前記ページディレクトリー及び前記複数のテーブルの各々は、前記メモリの１または複数のページに格納され、前記マップの前記表現は、前記１または複数のページの内の第１のページに基づく少なくとも１つのシャドウページを含み、前記シャドウページは、前記ページディレクトリーまたは前記複数のテーブルの内の１つの代替のバージョンを含み、前記マップは、前記仮想アドレスに基づいて前記メモリにアクセスするために使用される場合に、アクセスが許可されるページを定義するメモリアクセスポリシーに基づいてアクセスが許可されない少なくとも１つの態様を含み、前記シャドウページは、前記仮想アドレスに基づいて前記メモリにアクセスするための前記マップの前記表現が前記メモリアクセスポリシーに基づいてアクセスが許可されるような方法で、前記１または複数のページの内の前記第１のページと異なり、前記１または複数のページの内の前記第１のページを使用するソフトウェアオブジェクトは、前記ソフトウェアオブジェクトの現在の動作に関して、前記ソフトウェアオブジェクトが前記複数のページの内の第１のページに以前に書き込んだデータであって、前記複数のページの内の前記第１のページ内に存在するデータに依存する非アドレスマッピング動作を実行し、前記データは、前記複数のページの内の前記第１のページ内に存在するが、前記シャドウページ内に存在しない特性を有し、前記シャドウページは、前記ソフトウェアオブジェクトに関するアドレスを変換するために用いられ、前記複数のページの内の前記第１のページは、前記ソフトウェアオブジェクトに関するアドレスを変換するために用いられない、ステップと、

前記アクセスリクエストで特定された読み出しまたは書き込みを実行するステップと

を備えることを特徴とする方法。

【請求項１３】

前記メモリアクセスポリシーは、アクセス不可として前記メモリの一部を定義し、前記マップは、仮想アドレスマッピングを定義する前記メモリの部分への書き込み可能なリンクを公開し、前記マップの前記表現は、仮想アドレスマッピングを定義する前記メモリの部分への書き込み可能なリンクを公開しないことを特徴とする請求項１２に記載の方法。

【請求項１４】

前記複数のテーブルは、前記１または複数のページの組へのリンクを含み、前記ページディレクトリーは、前記複数のテーブルへのリンクを含み、前記少なくとも１つのシャドウページは、前記シャドウディレクトリーの中の少なくとも１つのリンクが、少なくとも前記複数のテーブルの１つの代わりにシャドウページテーブルを指すという点で前記ディレクトリーとは異なるシャドウディレクトリーを含むことを特徴とする請求項１２に記載の方法。

【請求項１５】

前記複数のテーブルは、前記１または複数のページの組へのリンクを含み、前記シャドウページは、前記テーブルの１つに基づく表現を含み、前記シャドウページは、前記テーブルの前記１つに存在する第１のリンクの表現を含み、前記第１のリンクは、前記テーブルの前記１つの中の読み出し／書き込みリンクであり、前記シャドウページは、前記シャドウページ表現の前記第１のリンクが読み出し専用にマークされている点で、前記テーブルの前記１つと異なることを特徴とする請求項１２に記載の方法。

【請求項１６】

前記シャドウページは、ディレクトリーを含み、前記メモリユニットは、複数の第２サイズのページを含む第１サイズのページによって網羅され、前記マップは、前記第１サイズのページへのリンクを含むディレクトリーを含み、前記シャドウページは、前記ディレクトリーに基づいており、前記シャドウページは、前記シャドウページが前記第１サイズのページへのリンクの代わりにテーブルへのリンクを含むという点で、前記ディレクトリーとは異なり、前記テーブルは、前記第１サイズのページに含まれる第２サイズのページへのリンクを含むことを特徴とする請求項１２に記載の方法。

【請求項１７】

アドレス変換マップを表すデータ構造を格納したコンピュータ読取り可能な記録媒体であって、前記アドレス変換マップは、ページディレクトリーを含み、前記ページディレク

トリーは、複数のページテーブルへのリンクを含み、前記複数のページテーブルの各々は、前記コンピュータ読取り可能な記録媒体の中の特定のフレームに格納され、前記複数のページテーブルの各々は、前記コンピュータ読取り可能な記録媒体の複数のページへのリンクを含み、前記データ構造は、複数のページテーブルの内の第１のページテーブルに基づくシャドウページテーブルと、前記ページディレクトリーに基づくシャドウページディレクトリーとを含み、前記ページディレクトリーは、前記複数のページテーブルの内の前記第１のページテーブルへのリンクを含む第１のエントリーを含み、前記シャドウページテーブルは、前記第１のエントリーに対応する第２のエントリーを含み、前記第２のエントリーは、前記複数のページテーブルの内の前記第１のページテーブルへのリンクの代わりに前記シャドウページテーブルへのリンクを含み、前記複数のページテーブルの内の前記第１のページテーブルを使用するソフトウェアオブジェクトは、前記ソフトウェアオブジェクトが前記複数のページテーブルの内の前記第１のページテーブルに以前に書き込んだデータであって、前記複数のページテーブルの内の前記第１のページテーブルに存在するデータに依存する非アドレスマッピング動作を実行し、前記データは、前記複数のページテーブルの内の前記第１のページテーブル内に存在するが、前記シャドウページテーブル内に存在しない特性を有し、前記シャドウページテーブルは、前記ソフトウェアオブジェクトに関するアドレスを変換するために用いられ、前記複数のページテーブルの内の前記第１のページテーブルは、前記ソフトウェアオブジェクトに関するアドレスを変換するために用いられないことを特徴とするコンピュータ読取り可能な記録媒体。

【請求項１８】

前記複数のページテーブルの内の前記第１のページテーブルは、第１のフレームに格納され、前記シャドウページテーブルは、第２のフレームに格納され、前記ページディレクトリー中のリンクが、前記第１のフレームの識別子を含み、および、前記シャドウページディレクトリー中の対応するリンクが、前記第２のフレームの識別子を含む点において、前記シャドウページディレクトリーは、前記ページディレクトリーと異なることを特徴とする請求項１７に記載のコンピュータ読取り可能な記録媒体。

【請求項１９】

前記複数のページテーブルの前記第１のページテーブルは、前記複数のページの内の第１のページへのリンクを含み、前記シャドウページテーブルは、前記複数のページの内の前記第１のページへのリンクの代わりに該第１のページに基づく表現へのリンクを含み、前記第１のページに基づく前記表現は、前記第１のページとは異なるフレームに格納されることを特徴とする請求項１７に記載のコンピュータ読取り可能な記録媒体。

【請求項２０】

前記複数のページの内の前記第１のページは、前記ページディレクトリーまたは前記複数のページテーブルの内の前記第１のページテーブルのいずれかを格納することを特徴とする請求項１９に記載のコンピュータ読取り可能な記録媒体。

【請求項２１】

前記複数のページテーブルの内の前記第１のページテーブルは、読み出し可能かつ書き込み可能として前記複数のページの内の前記第１のページを指定するリンクを含み、前記シャドウページテーブル中の対応するリンクは、前記複数のページの内の前記第１のページを読み出し可能のみとして指定することを特徴とする請求項２０に記載のコンピュータ読取り可能な記録媒体。

【請求項２２】

前記ページディレクトリーおよび前記複数のページテーブルの内の前記第１のページテーブルは、仮想アドレスに基づいてメモリにアクセスするために使用される場合、アクセスが許可されるページを定義するメモリアクセスポリシーに基づいてアクセスが許可されないような少なくとも１つの特徴を含み、前記シャドウページディレクトリーおよび前記シャドウページテーブルは、前記仮想アドレスに基づいて前記シャドウページディレクトリーおよび前記シャドウページテーブルを通したメモリへのアクセスが、前記メモリアクセスポリシーに基づいて許可されるようなデータを含むことを特徴とする請求項１７に記

載のコンピュータ読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、一般に、コンピュータのメモリ管理の分野に関し、特に、仮想アドレスシステムにおけるアドレス変換テーブルの管理に関する。
【背景技術】
【０００２】
現代のコンピュータシステムは、一般に、ある種の仮想アドレスメカニズムを備えている。この技術分野で周知のように、コンピュータシステムに関連付けられそれぞれが個々にアクセスできるメモリユニットは、そのメモリユニットを一意に識別する物理アドレスをもっている。しかしながら、仮想アドレス付けをサポートするコンピュータシステムでは、仮想アドレスを物理アドレスに割り当てることが可能である。仮想アドレスシステムは、仮想アドレスを物理アドレスに変換するための変換マップを使用する。
【０００３】
仮想アドレスシステムの一つの特徴は、物理アドレスの特定の組（物理メモリのページなど）が、いかなる仮想アドレスをももたないようなアドレス変換マップを構成することができるということである。典型的なページベースのメモリ管理スキームでは、アドレス変換マップは、仮想ページ記述子を物理ページフレーム番号に変換する。したがって、アドレス変換マップが、所与の物理ページフレームに導かないことを保証することによって、仮想アドレスに、そのページフレーム内のすべての位置を与えないことができる。さらに一般的には、多くの仮想アドレッシングスキームは、仮想アドレスを介して実行することができるアクセス（リード、リード／ライトなど）で、仮想アドレスにタグを付ける。あるページへの選択されたアクセス（書き込みなど）は、そのページへのどの仮想アドレスマッピングも、拒否されたアクセスを許可しないことを保証することによって妨ぐことができる。アドレス変換マップのこの面を使用して、メモリ保護の方法を実装することができる。したがって、ソフトウェアオブジェクト（オペレーティングシステム、アプリケーションレベルの処理、または他のいかなる種類のソフトウェアオブジェクトなど）による、物理アドレス空間のページへのアクセスを、このソフトウェアオブジェクトに公開されたどのマップも、問題のページへのどの仮想アドレスマッピングもアクセスを許可しないような状態にすることを保障することによって、拒否することができる。この種のメモリ保護スキームは、特にＩＡ３２ファミリーのプロセッサ（インテル（登録商標）ｘ８６プロセッサなど）において有用である。なぜならば、インテル（登録商標）ｘ８６プロセッサのアーキテクチャは、プロテクトモード（このプロセッサの通常の動作状態）で動作しているとき、すべてのメモリアクセスリクエストは、仮想アドレス変換を通るようになっているからである。特定の物理アドレスへの特定のアクセスを許す方法で、スーパーバイザーモードプログラムが、変換テーブルを変更することを禁止することによって機能するメモリプロテクションスキームは、“アドレス変換制御”、またはＡＴＣと呼ばれる。
【０００４】
典型的なアーキテクチャ（ｘ８６など）においては、仮想アドレスから物理アドレスへの変換は、通常のメモリページ（“ページマップ”ページと呼ばれる）の中身により与えられる。これは、書き込みオペレーティングシステムのために好都合である。なぜなら、仮想アドレスマップは、通常のメモリオペレーションによって作成し変更することができるからである。もし、オペレーティングシステムがＡＴＣを使用することを制限されることになっている場合、ＡＴＣは、オペレーティングシステムが、ページマップページに直接書き込むことを許すマッピングをもつことを防がなければならない。というのは、オペレーティングシステムは、任意の物理メモリページへの任意のアクセスをそれに与えるマッピングを作成するためのそのようなページへの書き込みを使用することができるからである。したがって、ソフトウェアオブジェクトが書き込みを許されないページへの読み出し／書き込みマッピングを防ぐことに加えて、ＡＴＣは、ページマップページへの読み出

し／書き込みマッピングを含む「危険な」マップを妨げなければならない。

【０００５】

ＡＴＣによるメモリアイソレーションが有効である間、発生する１つの問題は、危険なマップを作成するが、それ自身はアクセス制御ポリシーに違反しない書き込みリクエストをどのように扱うかである。このような書き込みリクエストを扱う１つの方法は、簡単には、このリクエストを失敗させることである。しかしながら、これは、オペレーティングシステムに実質的な改訂が必要となる。したがって、いくつかの現在のＡＴＣアルゴリズムは、書き込まれた値を変更するか（例えば、ページマップページへの読み出し・書き込みマッピングを読み出し専用マッピングに変更するなど）またはマップを安全に作成するために他のページマップページを変更する。この技術に関する問題は、実際には、ターゲットの位置は結局異なる値を含むのに、特定の値がターゲットの位置に書き込まれていると信じて、ソフトウェアオブジェクトが書き込みリクエストを実行することである。この不一致は、様々な方法で影響を及ぼす可能性がある。例えば、ソフトウェアオブジェクトは、このソフトウェアが格納したと思っている値に基づくチェックサムを生成する場合がある。そして、これらのチェックサムは、ＡＴＣシステムによって生成された修正値に対し無効となる。

【発明の開示】

【発明が解決しようとする課題】

【０００６】

本発明の一実施の形態の利点として、危険なマップを作成する（しかし、セキュリティポリシーに従う）書き込みが、変更されずに成功するように見えるが（ソフトウェアオブジェクトの見地から）、セキュリティポリシーを回避するために結果のマップを利用することができないようにすることよって、従来技術の欠点を克服する環境を提供する。

【課題を解決するための手段】

【０００７】

本発明は、シャドウページの使用が、アドレス変換制御を支援することを提供する。典型的な仮想アドレッシングでは、所与のページは、マップページ（マップの一部であるデータを含む）またはデータページ（ある仮想アドレスのターゲット）か、あるいは両方でありうる。マップページとデータページは、異なる状況でアクセスされる。データページのエントリーは、基礎となっている読み出しリクエストまたは書き込みリクエストのターゲットである。マップページのエントリーは、他方、他のページの場所を見つけるために順に逆参照される。本発明は、（例えば、以下に説明されるように、ディレクトリー、テーブル、またはデータページとして）ページが使用されうる異なる状況に対応するページの複数のコピーを保持する。本発明は、ページがアクセスされる状況に依存して、ページの適切なコピーを使用する。

【０００８】

いくつかの仮想アドレッシングシステム（インテル（登録商標）ｘ８６ファミリーのプロセッサで用いられる最も一般的な仮想アドレスモードなど）は、２つの種類のマップページ、すなわちディレクトリーおよびテーブルを持っている。ディレクトリーは、テーブルおよびラージデータページへの参照を含み、テーブルは、スモールデータページへの参照を含んでいる。（“ラージ”および“スモール”ページについては、以下にてさらに詳細に説明する。）したがって、アドレス変換処理の観点から、所与のページがディレクトリーとして、テーブルとして、またはターゲットデータとして、アクセスされうる３つまでの異なる状況がある。好ましい実施形態では、所与のページの３つまでのバージョン、すなわち、ディレクトリーバージョン、テーブルバージョン、およびデータバージョン、が保持される。所与のページがアクセスされると、そのページのディレクトリーバージョン、テーブルバージョン、またはデータバージョンが、ページがアクセスされている状況に依存して使用される。

【０００９】

ＡＴＣのもとでは、ページの中身は、ページがディレクトリーまたはテーブルとして使

用される場合のみ、メモリアクセスポリシーの違反を引き起こす場合がある。例えば、ページは、立入禁止ページへのリンクを含むことができる。しかしながら、このページが立入禁止ページにアクセスするために使用される危険は、アドレス変換器が実際にマップの一部としてこのページを使用している場合に存在するだけである。もし、このページが、代わりに、データページとしてアクセスされているなら、それは立入禁止ページに対する仮想アドレスを見せない。したがって、このページのデータコピーは、ソフトウェアオブジェクトがそのページに書き込んだと信じる実際のデータを含む場合がある。一方、そのページのディレクトリーおよびテーブルのコピーは、安全なマップを与える変更されたバージョンを含む場合がある。

【００１０】

本発明の他の特徴は、以下に記述される。

添付の図面とともに、以上の課題を解決するための手段と後述の発明を実施するための最良の形態を読むと、本発明をよりよく理解される。本発明を説明するために本発明の例示的な構成を図示するが、本発明は開示するこの特定の方法および手段には限定されない。

【発明を実施するための最良の形態】

【００１１】

（概要）

アドレス変換制御を使用して、メモリアクセス制御ポリシーに違反して使用されうる仮想アドレスマッピングを効果的に拒否することによって、メモリアクセス制御ポリシーを実装することができる。一般的に、アドレス変換制御は、アドレス変換マップを編集する際の試行に実質的な制限をかけることによって、このマップが安全なままであるように（このマップが、所与のソフトウェアのエンティティに対し、このエンティティによって立入禁止（または書き込み不可）となっているページへのリンク（または、書き換え可能なリンク）を見えないようにするという意味で）有効に働く。典型的には、マップを編集するためのリクエストを実行することが、このマップを望ましくない状態に置くかどうか判断するために、これらの実質的な制限が、このリクエストを評価することによって課せられる。望ましくない状態が起きる場合、要求されたリクエストを実行することが、要求された状態を維持するように、このリクエストは変更される。（例えば、結果的に、このポリシーの下で読み出し可能だが書き込み不可であるページへの読み出し／書き込みリンクとなるマップを編集するためのリクエストを、読み出し専用としてこのリンクをマークするように変更することができる。）この技術に関する問題は、ソフトウェアの正しい振る舞いが、ときには、このソフトウェアがメモリに書き込んだと信じる値を含んでいるメモリに依存するということである。（例えば、チェックサムをヴェリファイする場合では）リクエストを変更することは、ソフトウェアが書き込んだと信じる値とは異なる値をメモリに含ませる原因となる。本発明は、ページマップのページとして使用するページの異なる複数のバージョン、すなわち、ソフトウェアオブジェクトにさらされたデータバージョン、および、マップの安全性を壊さずに、アドレス変換処理の一部として使用することができる１または複数のマップバージョン、を保持することによってこの問題に対処する。このようなページのデータバージョンへのマップは、読み出し専用にされる。結果、このページへの書き込みは、異なるバージョンが同期するようにページを編集することができるＡＴＣによって遮られる。

【００１２】

（例示的なコンピューティング環境）

図１は、本発明の諸態様を実施することができる例示的なコンピューティング環境を示している。コンピューティングシステム環境１００は適切なコンピューティング環境の一例にすぎず、本発明の使用または機能の範囲に関するいかなる限定を示唆することも意図したものでもない。また、このコンピューティング環境１００は、例示的オペレーティング環境１００の中に示すいかなるコンポーネントの１つまたは組合せに関し、なんらかの依存性も要件ももつとは解釈されるべきではない。

【００１３】
本発明は、他の多くの汎用または専用のコンピューティングシステム環境または構成にも適用することができる。本発明の使用に適した周知のコンピューティングシステム、環境、および／または構成の例には、限定するものではないが、パーソナルコンピュータ、サーバコンピュータ、ハンドヘルドまたはラップトップ装置、マルチプロセッサシステム、マイクロプロセッサベースのシステム、セットトップボックス、プログラム可能な家庭用電化製品、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、埋め込みシステム、上記の任意のシステムまたは装置を含む分散コンピューティング環境などが含まれる。
【００１４】
本発明は、プログラムモジュールのようにコンピュータ実行可能命令をコンピュータが実行する一般的なコンテキスト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、データ構造などがあり、特定のタスクを実行するものや、特定の抽象データ型を実装するものがある。本発明は、通信ネットワークや他のデータ送信媒体を介してリンクするリモート処理装置がタスクを実行する分散コンピューティング環境でも実施できる。分散コンピューティング環境では、メモリ記憶装置を含むローカルとリモートの両方のコンピュータ記憶媒体にプログラムモジュールとそれ以外のデータを置くことができる。
【００１５】
図１を参照すると、本発明を実施する例示的なシステムは、コンピュータ１１０の形で汎用コンピューティング装置を含んでいる。コンピュータ１１０の構成要素には、限定するものではないが、処理装置１２０、システムメモリ１３０、および、システムメモリの処理装置１２０への接続を含む様々なシステム構成部品を接続するシステムバス１２１が挙げられる。処理装置１２０は、マルチスレッドプロセッサ上でサポートされるものなど、複数の論理処理装置を代表するものとすることができる。システムバス１２１は、様々なバスアーキテクチャのいずれかを使用するメモリバスまたはメモリコントローラ、周辺バス、およびローカルバスを含む各種バス構造のいずれでもよい。限定するものではないが、こうしたアーキテクチャには、例として、ＩＳＡ（Ｉｎｄｕｓｔｒｙ Ｓｔａｎｄａｒｄ Ａｒｃｈｉｔｅｃｔｕｒｅ）バス、ＭＣＡ（Ｍｉｃｒｏ Ｃｈａｎｎｅｌ Ａｒｃｈｉｔｅｃｔｕｒｅ）バス、ＥＩＳＡ（Ｅｎｈａｎｃｅｄ ＩＳＡ）バス、ＶＥＳＡ（Ｖｉｄｅｏ Ｅｌｅｃｔｒｏｎｉｃｓ Ｓｔａｎｄａｒｄｓ Ａｓｓｏｃｉａｔｉｏｎ）ローカルバス、ＰＣＩ（Ｐｅｒｉｐｈｅｒａｌ Ｃｏｍｐｏｎｅｎｔ Ｉｎｔｅｒｃｏｎｎｅｃｔ）バス（メザニン（Ｍｅｚｚａｎｉｎｅ）バスとしても知られている）が挙げられる。システムバス１２１は、通信装置間の、ポイントツーポイント接続、スイッチングファブリック（ｓｗｉｔｃｈｉｎｇ ｆａｂｒｉｃ）または同様のものとしても実現できる。
【００１６】
コンピュータ１１０には、通常は様々なコンピュータ読取可能な媒体が含まれる。コンピュータ読取可能な媒体は、コンピュータ１１０からアクセスできる任意の使用可能な媒体でよく、揮発性と不揮発性の両方、および取り外し可能と固定の両方の媒体を含む。限定するものではないが、例として、コンピュータ読取可能な媒体には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は、コンピュータ読取可能な命令、データ構造、プログラムモジュール、またはその他のデータなどの情報の記憶のための任意の方法または技術で実装された、揮発性と不揮発性の両方、および取り外し可能と固定の両方の媒体が含まれる。コンピュータ記憶媒体には、限定するものではないが、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、フラッシュメモリなどのメモリ技術、ＣＤ ＲＯＭ、デジタル多用途ディスク（ＤＶＤ：ｄｉｇｉｔａｌ ｖｅｒｓａｔｉｌｅ ｄｉｓｋｓ）などの光ディスク記憶装置、磁気カセット、磁気テープ、磁気ディスクなどの磁気記憶装置、または必要な情報を格納するのに使用することができ、コンピュータ１１０からアクセスできる他の任意の媒体が挙げられる。通信媒体は、典型的には、搬送波やその他

の搬送メカニズムなどの変調されたデータ信号中のコンピュータ読取可能な命令、データ構造、プログラムモジュール、またはその他のデータなどを具現化するものであり、任意の情報伝達媒体を含む。「変調されたデータ信号」という用語は、信号内に情報を符号化するような方法で、１つまたは複数の特性が設定または変更された信号を意味する。限定するものではないが、通信媒体には、例として、有線ネットワーク、直接ワイヤ接続などの有線媒体と、音響、無線、赤外線などの無線媒体が挙げられる。上記の任意の組合せも、コンピュータ読取可能な媒体の範囲内に含まれるものとする。

【００１７】

システムメモリ１３０には、読み取り専用メモリ（ＲＯＭ：ｒｅａｄ ｏｎｌｙ ｍｅｍｏｒｙ）１３１やランダムアクセスメモリ（ＲＡＭ：ｒａｎｄｏｍ ａｃｃｅｓｓ ｍｅｍｏｒｙ）１３２などの揮発性および／または不揮発性のメモリという形をとるコンピュータ記憶媒体が含まれる。起動時などにコンピュータ１１０内の構成要素間の情報転送を支援する基本ルーチンを含む基本入出力システム１３３（ＢＩＯＳ：ｂａｓｉｃ ｉｎｐｕｔ／ｏｕｔｐｕｔ ｓｙｓｔｅｍ）は、通常はＲＯＭ １３１に格納される。ＲＡＭ １３２には、通常処理装置１２０から直ちにアクセスできる、かつ／または処理装置１２０で現在操作しているデータおよび／またはプログラムモジュールが入っている。限定するものではないが、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、その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。

【００１８】

コンピュータ１１０には、その他の取り外し可能／固定、揮発性／不揮発性のコンピュータ記憶媒体を含めてもよい。単なる一例として、図１に、固定された不揮発性の磁気媒体の読み出しまたは書き込みを行うハードディスクドライブ１４０、取り外し可能な不揮発性の磁気ディスク１５２の読み出しまたは書き込みを行う磁気ディスクドライブ１５１、ＣＤ ＲＯＭや他の光媒体などの取り外し可能な不揮発性の光ディスク１５６の読み出しまたは書き込みを行う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することが可能な上記以外の取り外し可能／固定、揮発性／不揮発性のコンピュータ記憶媒体には、限定するものではないが、磁気テープカセット、フラッシュメモリカード、ＤＶＤ、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などが挙げられる。ハードディスクドライブ１４１は、通常インターフェース１４０などの固定のメモリインターフェースを介してシステムバス１２１に接続し、磁気ディスクドライブ１５１と光ディスクドライブ１５５は、通常インターフェース１５０などの取り外し可能なメモリインターフェースを介してシステムバス１２１に接続する。

【００１９】

図１に示す上述のドライブとこれに対応するコンピュータ記憶媒体には、コンピュータ１１０に対するコンピュータ読取可能な命令、データ構造、プログラムモジュールおよびその他のデータの記憶を提供する。例えば、図１では、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、その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を記憶しているように描かれている。こうした構成部品は、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、その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と同じでも、異なっていてもよいことに留意されたい。ここでは、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、その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には異なる番号を付けて、少なくとも別の複製であることを示している。ユーザは、キーボード１６２やポインティングデバイス１６１（一般にマウス、トラックボール、またはタッチパッドと呼ばれる）を介してコンピュータ２０にコマンドや情報を入力できる。他の入力装置（図示せず）には、マイクロフォン、ジョイスティック、ゲームパッド、サテライトディッシュ（ｓａｔｅｌｌｉｔｅ ｄｉｓｈ）、スキャナなどが挙げられる。これらの入力装置および他の入力装置は、多くの場合システムバスに接続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に接続するが、

パラレルポート、ゲームポート、ＵＳＢ（ｕｎｉｖｅｒｓａｌ　ｓｅｒｉａｌ　ｂｕｓ）のような他のインターフェースやバス構造によって接続してもよい。モニタ１９１または他の種類の表示装置も、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。モニタに加えて、コンピュータはスピーカー１９７やプリンタ１９６などその他の周辺出力装置も含むことができ、これらは出力周辺インターフェース１９５を介して接続することができる。

【００２０】

コンピュータ１１０は、リモートコンピュータ１８０などの１台または複数台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、ピアデバイス（ｐｅｅｒ　ｄｅｖｉｃｅ）、または他の一般のネットワークノードでよく、通常は、コンピュータ１１０に関連して上述した構成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のみを示す。図１に示す論理接続には、ローカルエリアネットワーク（ＬＡＮ：ｌｏｃａｌ　ａｒｅａ　ｎｅｔｗｏｒｋ）１７１とワイドエリアネットワーク（ＷＡＮ：ｗｉｄｅ　ａｒｅａ　ｎｅｔｗｏｒｋ）１７３とが含まれるが、他のネットワークを含めてもよい。このようなネットワーキング環境は、職場、企業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はごく一般的である。

【００２１】

ＬＡＮネットワーキング環境で使用される場合、コンピュータ１１０はＬＡＮ　１７１にネットワークインターフェースまたはアダプタ１７０を介して接続する。ＷＡＮネットワーキング環境で使用される場合、コンピュータ１１０は通常インターネットなどのＷＡＮ　１７３を介して通信を確立するためのモデム１７２またはその他の手段を備えている。内蔵または外付けすることができるモデム１７２は、ユーザ入力インターフェース１６０または他の適切なメカニズムを介してシステムバス１２１に接続できる。ネットワーク環境では、コンピュータ１１０またはその一部に関連して示したプログラムモジュールは、リモートメモリ記憶装置に格納することができる。限定するものではないが、例として、図１にリモートアプリケーションプログラム１８５がメモリデバイス１８１にあるものとして示す。当然のことながら、図示されたネットワーク接続が例示的なものであり、コンピュータ間の通信リンクを確立する他の手段を使用してもよいことは言うまでもない。

【００２２】

（仮想アドレススキームの例）

図２は、仮想アドレスシステムの一例を示している。図２に描かれた例は、ページ型の仮想アドレススキームである。ただしこれは、セグメンテーションのように、仮想アドレッシングが他のモデルに基づくことができることを理解されるであろう。図２に示すスキームは、インテル（登録商標）ｘ８６プロセッサ上で利用できる仮想アドレッシングスキームの１つなどの２レベルのアドレススキームである。このスキームは、下記のように、１つが、仮想ページ識別子を物理ページに変換するために、２つのレベルの手段を使用しなければならないという意味で“２レベル”である。

【００２３】

このページングスキームでは、ページディレクトリー２０２は、エントリーの１組（ｓｅｔ）を含んでいる。以下に、エントリーの一例の構造を図３と関係付けて、さらに詳細に説明する。しかし、本質的には、各エントリーは、ページテーブル２０４（１）、２０４（２）または２０４（３）などの、特定のページテーブルの物理的位置（すなわち、ページフレームナンバーまたは“ＰＥＮ”）を特定する。各ページテーブルも同様に、エントリーの組を含んでいる。ここでは、各エントリーは、ページ２０６（１）、２０６（２）、２０６（３）、２０６（４）などの、物理的位置（同様に、ページフレームナンバー）と特定のデータページを同一視する。データページは、ＲＡＭ１３２の予め定義された長さの隣接する部分にある。データページは、任意の種類のデータを格納することができる。そして、注目すべきは、通常のデータの格納に加え、データページはまた、ページデ

ィレクトリー２０２およびページ２０４（１）から２０４（３）までの中身を格納するのにも使用されることに留意されたい。したがって、所与のページは、ディレクトリー、テーブル、データページであることができ、または、これらの３つの構造の任意の組み合わせと同じように複数の役割を演じることができる。

【００２４】

図２に描かれた仮想アドレススキームは、２レベル仮想アドレススキームである。これは、特定のページを配置するために、ページディレクトリー（レベル１）とページテーブル（レベル２）の両方を使用する必要があるからである。任意のレベル番号を用いて仮想アドレスシステムを設計することができ、そして、本発明の原理は、このようなすべての仮想アドレススキームに適用することができることを、当業者は理解されよう。この分野で周知のように、インテル（登録商標）ｘ８６プロセッサは、１または２または３レベルを有する仮想アドレスをサポートしており、典型的には、“ハイブリッド”スキームを使用する。このスキームでは、“スモール”ページ（すなわち、４キロバイト長であるページ）は、２レベル仮想アドレスを使用し、一方、“ラージ”ページ（すなわち、４メガバイト長であるページ）は、１レベル仮想アドレスを使用する。

【００２５】

図２のページングスキームでは、ページ上のどのバイトも、ページディレクトリーオフセット２１１と、ページテーブルオフセット２１２と、ページオフセット２１３とを含む仮想アドレス２１０によって特定することができる。（仮想アドレスのこの構造は、スモールページに格納されるデータに適用される。ラージページについては以下に説明する。）したがって、物理アドレスを配置するために、アドレス変換を実行するメモリマネジメントユニット（ＭＭＵ）２２０が、ページディレクトリー２０２内の特定のエントリーに配置するために、ページディレクトリーオフセット２１１を使用する。例えば、オフセット２１１は、ゼロに等しくして、ページディレクトリー２０２のゼロ番目のエントリーが、調べられるべきであるということを示すことができる。このエントリーは、ページテーブルが格納されるＰＦＮを含んでおり、そこでＭＭＵ２２０は、このＰＦＮをページテーブルの１つ（ページテーブル２０４（１）など）を見つけるために使用する。ＭＭＵ２２０は、このときページテーブルオフセット２１２を、特定されたページテーブルへのインデックスとして使用し、そのオフセットで見つけられるエントリーを検索する。このエントリーは、データページ（ページ２０６（１）など）のＰＦＮを含む。そのためＭＭＵ２２０は、物理メモリの特定のバイトを見つけるために、特定されたページのベースアドレスにページオフセット２１３を加算する。ＭＭＵ２２０は、単なるアドレス変換に加えて、他の様々な機能を実行することにも適応することができる。例えば、ＭＭＵ２２０は、テーブル中のページのエントリーが“非存在”とマークされている場合には、ディスクからそのページをロードすることができる。また、ＭＭＵ２２０は、“読み出し専用”などとマークされている場合には、書き込みアクセスを禁止することができる。

【００２６】

もし、仮想アドレスがラージページを参照するなら、仮想アドレスの構造およびそのアドレスを変換する処理は、上述したものとはわずかに異なる。仮想アドレスは、唯一のオフセットを含み、このオフセットは、ディレクトリーへのインデックスである。このオフセットに配置されたディレクトリーエントリーは、ページテーブルのＰＦＮを含む代わりに、ラージデータページのＰＦＮを含む。ディレクトリーエントリーはまた、エントリーが、ページテーブルの代わりにラージページを参照することを示すために設定される１ビットを有している。ラージページビットが設定される場合、仮想アドレスは、ページテーブルへのインデックスを含まない。そのため、どのページテーブルも変換処理において使用されない。代わりに、仮想アドレスの残りの部分（すなわち、ディレクトリーへのインデックス以外の部分）が、ラージページへのインデックスとして扱われる。このページテーブルのレベルはバイパスされ、それで唯一の変換レベルが起こる。

【００２７】

図２の仮想アドレススキームにおいては、ページディレクトリー自身の位置（例えば、

ＰＦＮ）が、記憶位置２０１に格納される。ＭＭＵ２２０は、仮想アドレス２１０の変換を開始すると、ページディレクトリー２０２を配置するために、この記憶位置の中身を使用する。したがって、既存の複数のページマップが存在することができる。そして、所与のマップのページディレクトリーのＰＦＮを含むように記憶位置２０１の中身を設定することによって、現在の使用のために特定のマップを選択することができる。インテル（登録商標）ｘ８６プロセッサを例とすると、記憶位置２０１は、ＣＲ３と命名されたレジスタに対応する。

【００２８】

上述のように、ページテーブルまたはページディレクトリー中の各エントリーは、特定の物理ページのＰＦＮを含んでおり、また、各エントリーは、特定の他のデータを含んでいてもよい。図３は、ページテーブルまたはページディレクトリー中のエントリー３００の構造例を示している。

【００２９】

エントリー３００は、特定の物理ページのＰＦＮ３０２を含んでいる。例えば、エントリー３００が、ページディレクトリーの一部である場合、ＰＦＮ３０２は、アドレス変換処理の次のレベルにおいて参照されるべきページテーブルのＰＦＮである（または、ラージページリンクの場合では、このエントリーは単に、このエントリーが参照するラージデータページのＰＦＮを含むものである）。さらに、エントリー３００は、このエントリーがラージページのためのものであるかスモールページのためのものであるかを示すビット３０４を含む。（このビットは、エントリー３００がページディレクトリーの一部分であるときのみ意味をもつ。簡単のため、このエントリーのフォーマットは、テーブルの場合では、ビット３０４の意味は未定義だが、このエントリーがディレクトリーの一部であろうとテーブルの一部であろうと同じにすることができる。）

読み出し専用のビット３０６は、エントリーの最終のターゲットであるデータページが、読み出し／書き込み（この場合、ビットはクリアされている）として扱われるべきであるかまたは読み出し専用（この場合、ビットはセットされている）として扱われるべきであるかを示す。もし、ターゲットのデータページが読み出し専用である場合、そのページへの書き込みリクエストは、失敗する。（ＭＭＵ２２０を使用して、ページの読み出し専用の状態を守らせることができる。）この読み出し専用ビットは、ディレクトリーのエントリーとテーブルのエントリーの両方に存在することができる。もし、データページへ最後に導かれるディレクトリーおよびテーブルのリンクが、各々の読み出し専用ビットに関し競合する設定となっている場合、競合解決ルールが使用され、ターゲットのデータページが読み出し／書き込みか読み出し専用かを判断することができる。例えば、競合ルールは、ページが読み出し／書き込みとして扱われるために、そのページへ導くディレクトリーのリンクおよびテーブルのリンクの両方が、読み出し／書き込みにマークされなければならない（両方のエントリー中の読み出し専用ビット３０６がクリアされなければならない）ことを提示することができる。同じページが、マップを通した異なるパスによって到達可能となることができる。そして、このページが読み出し／書き込みとして扱われるかまたは読み出し専用として扱われるかは、どのパスがそのページに到達するために使用されるかに依存するかもしれない。

【００３０】

存在（ｐｒｅｓｅｎｔ）ビット３０８は、ターゲットのデータページが、現在物理メモリに存在するか、ディスクからメモリにコピーされる必要があるかを示す。例えば、存在ビット３０８がクリアされる場合（ターゲットページが存在しないことを示す）、そのページに対するアクセスリクエストは、ページフォールトを生成するかもしれない。次いで、ディスクから物理メモリにそのページの内容をコピーするとともにアドレス変換マップをこのページの物理的位置に反映させるためにアドレス変換マップを適合させる割込みサービスルーチンによって、このアクセスリクエストは処理される。存在ビットが、所与のマッピングのために、ページディレクトリーおよびページテーブルのエントリーに異なって設定されている場合、これらのビット間の競合は、読み出し／書き込みビットついて上

述したものと同様の競合解決ルールによって解決することができる。（例えば、ディレクトリーのエントリーとテーブルのエントリーの両方が、存在するとマークされているときのみ、存在するものとしてそのマッピングを扱う。）

【００３１】

（アドレス変換テーブルのエントリー編集制御（ＡＴＣ）を使用するメモリアクセス制御）

図２から３に関連して説明した仮想アドレススキームの一つの特徴は、そこが、対応する仮想アドレスの存在しない物理メモリの一部であることが可能であることである。この所見の結果は、メモリのどの部分が与えられても、アドレス変換マップがメモリのその部分へ導かないことを保証することによって、メモリのその部分へのアクセスを禁止することができるということである。実際には、そのメモリ位置が仮想アドレスをもたないので、立入禁止区域にレンダリングされる（ｒｅｎｄｅｒｄ）。（多くのシステム（インテル（登録商標）ｘ８６プロセッサなど）では、ほとんどすべてのメモリアクセスリクエストは、仮想アドレスによって作成される。アクセスリクエストが、物理アドレスによって作成される制限された環境に対しては、パラレルアクセス制御メカニズムを使用することができる。）

【００３２】

ＡＴＣを使用して、メモリアクセス制御を達成する方法は、次の注釈の観点から説明される。ＮＡ（“ｎｏ　ａｃｃｅｓｓ”）は、あるポリシーの下で、アクセスが禁止されるページの組である。ＭＰ（“ｍａｐｐｅｄ　ｐａｇｅｓ”）は、アドレス変換マップによってアクセスできるページの組（すなわち、仮想アドレスが存在するページの組）である。条件ＮＡ∩ＭＰ＝φが真であることを保持し続ける限り、ＮＡのメンバーであるページへのアクセスは防ぐことができる。この条件は、図４に、ベン図（Ｖｅｎｎ　Ｄｉａｇｒａｍ）として描かれている。すなわち、ページ４０６は、マシン上で利用可能な物理ページの組である。ＭＰ４０２は、仮想アドレスが存在するページの組である。ＮＡ４０４は、このポリシーの下ではアクセスが許可されないページの組である。図４の条件が真であり続ける限り、このポリシーの下で立入禁止となっているページへアクセスするための仮想アドレスを使用することはできないであろう。なぜならば、マップは、これらのページへ導かない（すなわち、これらのページは仮想アドレスをもたない）からである。したがって、図４に描かれた条件を使用して、ＡＴＣを介したメモリアクセス制御を達成することができる。この条件は、“不変条件（ｉｎｖａｒｉａｎｔ）”と呼ばれる。というのは、ＡＴＣの目的が、この条件が真の状態から偽の状態へ変化することを妨げるために、アドレス変換マップへの変更を制限することにあるからである。

【００３３】

図４は、メモリアクセス制御のために使用される簡単な不変条件を表している。なお、図示する目的のためのみに示すものである。アクセス制御が実行されるべき環境に応じてもっと複雑な条件が可能である。例えば、どのエントリーがディレクトリーに（またはテーブルに）含まれるかについて明確なルールを設定することができる。そして、アクセス制御条件を守らせるのに役立つ読み出し専用および／または存在ビットを使用することによって、アクセス制御を洗練することができる。例えば、その使用により、インテル（登録商標）ｘ８６プロセッサ上のＡＴＣを介したメモリアクセス制御を達成することができるルールの組の一例を、以下に説明する。

【００３４】

Ｄ１は、ページディレクトリーとして使用することができるページの組である。Ｄ２は、ページテーブルとして使用することができるページの組である。Ｄ＝Ｄ１∪Ｄ２である「存在（ｐｒｅｓｅｎｔ）」とマークされた（すなわち、その存在ビットがセットされている）、ページディレクトリーまたはページテーブルの各エントリーは、“リンク（ｌｉｎｋ）”と呼ばれる。Ｄ１のあるページから、問題のＤ２のページへの小規模の読み出し－書き込みリンクが存在する場合、Ｄ２のページは、“書き込み有効（ｗｒｉｔｅ－ａｃｔｉｖｅ）”である。（「小規模の（ｓｍａｌｌ）」リンクは、一つのディレクトリーか

ら一つのテーブルへのリンクである（すなわち、最終的にスモールページへ導くディレクトリー中のリンクである。「大規模の（ｌａｒｇｅ）」リンクは、ラージページを指し示す一つのディレクトリー中のリンクである。）あるエンティティが読み出しおよび／または書き込みのアクセスを許可されるページを定義するポリシーが存在することを仮定している。

【００３５】

なお、以下の不変条件が主張される：

・ＣＲ３は、Ｄ１の中にある。

・すべてのＤ１とＤ２のページは、適切なポリシーの下で読み出し可能である。

・Ｄ１ページからのどのスモールリンクもＤ２ページを指す。

・Ｄ２ページからのリンクはこのポリシーの下で読み出し可能なページを指す。

・書き込み有効なＤ２ページからのどの読み出し－書き込みリンクも、このポリシーの下で書き込み可能であり、かつＤにはないページを指す。

・Ｄ１からのラージリンクのラージページのターゲットに含まれるどのスモールページも、このポリシーのもとで読み出し可能である。もし、このリンクが読み出し－書き込みである場合、スモールページもまた、このポリシーのもとで書き込み可能であってＤには存在しない。

【００３６】

ＡＴＣは、上記不変条件に違反するようなアドレス変換マップに対する変更を防ぐことを保証するために使用される。これらの不変条件を保持することは、問題のエンティティがこのポリシーに違反することができないことを保証する。

【００３７】

どの不変条件が課せられているかにかかわらず、リクエストが、実際には実行されない場合、最終的に不変条件が保持し続ける状態になるかどうか判断するために各アクセスリクエストを評価することによって、不変条件の真理を保持することができる。もし、結果の状態が不変条件を満足するなら、このリクエストは、実行される。しかしながら、もし不変条件が保持に失敗するなら、少なくとも２つのオプションがある。

（１）リクエストを拒否する。または、

（２）不変条件が満たされ続ける形式にリクエストを変更する。

オプション（１）は、実際には、多数のアクセスリクエストを拒絶する必要があるという不利がある。これは、コンピュータシステムの機能を崩壊させる。しかしながら、オプション（２）では、ソフトウェアオブジェクトは、１つの値を１つの記憶位置に書き込むことになる。この記憶位置は、ソフトウェアオブジェクトが書き込んだと信じている値と異なる変更された値を最後には記憶することになりうる。前述のように、ソフトウェアの補正機能（チェックサム検証など）は、ソフトウェアがメモリに書き込んだと信じる実際の値を格納しているメモリに依存しうる。したがって、オプション（２）もまた、ソフトウェアの機能を崩壊させる。本発明は、１つのページの複数のバージョンを格納することによって、この問題に対処する。１つのバージョンは、プログラムがそのページに書き込んでいると信じる正確なデータを含んでいる。このページの他のコピーは、もし、アドレス変換処理において使用される場合、保持されるべき適切な不変条件をもたらすデータのバージョンを含んでいる。

【００３８】

（シャドウページ）

本発明の一つの特徴によれば、ページの複数の表現が存在することができる。同じページの複数の表現は、プログラムがそのページに実際に書き込むデータを含むページのバージョン、およびページディレクトリーとページテーブルとして、アドレス変換処理において使用するのに安全である、ページの他の（「シャドウ（ｓｈａｄｏｗ）」）バージョンが存在することを保証する。この文脈において「使用するのに安全（Ｓａｆｅ ｔｏ ｕｓｅ）」とは、ディレクトリー（または、場合によってはテーブル）としてのシャドウページの使用が、ＡＴＣシステムによって適用される不変条件が違反されることを引き起こさ

ないであろうということを意味する。

【００３９】

好ましくは、ページｘが与えられると、そのページの３つのバージョンが存在する。これらは、ｄ（ｘ），ｔ（ｘ），およびｍ（ｘ）として参照される。ｄ（ｘ）は、ページの「ディレクトリー」バージョンである（すなわち、前述のアドレス変換処理におけるページディレクトリーとして使用されるのに適したページのバージョンである）。ｔ（ｘ）は、ページテーブルとして使用するのに安全であるページのバージョンである。ｍ（ｘ）は、ページの「メモリ」バージョンである（すなわち、１または複数のプログラムによってこのページに書き込まれた実際のデータを含むバージョンである）。本明細書の記述において、「ｄ（ｘ）」項は、ページｘのディレクトリーバージョンの内容か、ページｘのディレクトリーバージョンが格納されているＰＦＮを参照することができる。ｔ（ｘ）とｍ（ｘ）についても同様である。ｄ（ｘ），ｔ（ｘ），およびｍ（ｘ）項がページの内容を参照するか、またはそれのＰＦＮを参照するかは、状況から明らかになるか、または具体的に示されるであろう。

【００４０】

図５は、ｄ（ｘ），ｔ（ｘ），およびｍ（ｘ）がアドレス変換処理においてどのように使用されるかを示す。図５は、ｘ，ｙ，およびｚ（参照数字は、それぞれ５０２，５０４，および５０６）がラベル付けされた３つのページを参照する図である。ページｘはページディレクトリーであり、ページｙはページテーブルであり、そしてページｚはデータページである。注目すべきは、ページｘ，ｙ，およびｚは、複数の役割を演じることができるということである。したがって、ｘは、環境に依存して、ページディレクトリーかデータページとして機能することができる。ページｙは、ある状況ではページテーブルとして、他の状況ではページディレクトリーとして機能することができる。しかしながら、特定の仮想アドレスが変換されている図５の目的のために、および、そのアドレスを変換する目的のために、ページｘ，ｙ，およびｚがそれぞれディレクトリー、テーブル、およびデータページの役割を演じることを仮定している。

【００４１】

ページｘは、ｄ（ｘ）、ｔ（ｘ）、およびｍ（ｘ）バージョンに存在する。問題のアドレスを変換するために、ページｘ（つまりｄ（ｘ））のディレクトリーバージョンが調べられる。ディレクトリーとして、ｘのエントリーは、ターゲットのページテーブルのＰＦＮを含む。ｘとｄ（ｘ）間の根本的な差異は、ｄ（ｘ）がｔ（t_i）のＰＦＮを含む一方、ページディレクトリーｘにおける各ターゲットt_iに対し、ｘがt_iのＰＦＮを含むということである。（言い換えると、ｄ（ｘ）は、ターゲットページのオリジナルバージョンの代わりに、ターゲットページのテーブルバージョンを指すように変更されるということである）。

【００４２】

ページｄ（ｘ）が参照されるとき、ｄ（ｘ）の関連するエントリーは（すなわち、仮想アドレスのディレクトリーオフセット部によって指し示されるエントリー、図２の構成要素２１１）は、ページｔ（ｙ）を指し示す。次いで、ページｔ（ｙ）が、特定のデータページの場所を見つけるために調べられる。ｔ（ｙ）のエントリは、ターゲットデータページのＰＦＮを含む。ｔ（ｙ）およびｙ間の関係は、ｄ（ｘ）とｘとの関係と類似している。すなわち、ｙによって参照される各データページd_iに対し、ｔ（ｙ）がd_iのＰＦＮの代わりにｍ（d_i）のＰＦＮを含む。（しかしながら、注目すべきは、このページのオリジナルの位置にあるページのデータバージョンを格納することが、一般的に、メモリの最も効率的な使用となるので、ｍ（d_i）のＰＦＮが、一般にd_iのＰＦＮと同じになるということである。）仮想アドレスのテーブルオフセットフィールドによって指し示されるオフセット（例えば、図２の構成要素２１２）を使用して、テーブルｔ（ｙ）の適切なエントリーが配置される。そのエントリーは、特定のデータページのＰＦＮを参照し、この例では、ｍ（ｚ）である。

【００４３】

ページｍ（ｚ）が特定された後、ページｍ（ｚ）中のデータの適切なユニットが、仮想アドレスで指し示されるページオフセット（図２の構成要素２１３）に基づいて、アクセスされる。

【００４４】

したがって、従来のアドレス変換処理においては、データページへのパスがページｘから、ページｙへ、ページｚへと導く。本発明によるシャドウページテーブルが使用される場合、変換ページがページｄ（ｘ）から、ページｔ（ｙ）へ、ページｍ（ｚ）へと導く。

【００４５】

（ページｄ（ｘ），ｔ（ｘ），およびｍ（ｘ）の作成）

ページｄ（ｘ），ｔ（ｘ），およびｍ（ｘ）は、ページｘ上で定義された変換を実行することによって作成される。以下に、それらの変換の好ましい実施形態を説明する。

【００４６】

好ましくは、ｍ（ｘ）は、任意の種類の変更あるいはフィルタリングを受けずに、プログラムがページｘに書き込む実際のデータを表す。言い換えると、ｘからｍ（ｘ）への変換は、本質的には恒等変換である。

【００４７】

好ましくは、ｄ（ｘ）とｔ（ｘ）は、次の規則に従って作成される。存在とマークされているページｘのエントリーを参照したすべてのページt_iのために、ｄ（ｘ）の対応するエントリーが、t_iのＰＦＮの代わりにｔ（t_i）のＰＦＮを参照することを除いて、ｄ（ｘ）はｘと同じである。加えて、ターゲットページが、適切なポリシーの下で読み出し可能だが書き込み不可である場合、または、ターゲットページがページディレクトリーもしくはページテーブルである場合、そのエントリーは、読み出し専用にマークされる。

【００４８】

以下では、ｄ（ｘ）とｔ（ｘ）がどのようにして作成されるかの説明を形式的に行う。この説明の目的のために、Ｄ１は、ページディレクトリーとして使用可能であるＰＦＮの組である。そして、Ｄ２は、ページテーブルとして使用可能であるＰＦＮの組である。ステートメントＤ１．ｘは、ｘがＤ１のメンバーであることを意味し、Ｄ２．ｘは、ｘがＤ２のメンバーであることを意味する。メモリアクセス制御スキームが強制されるべきである適切なソフトウェアオブジェクトによって見られるように、Ｍをメモリマップであるとしよう。Ｍ．ｘ．ｅは、そのＰＦＮがｘである物理ページの第ｅエントリーに格納された値を参照する。Ｒ．ｘは、ｘが適切なポリシーの下で読み出し可能であることを意味し、Ｗ．ｘは、ｘが適切なポリシーの下で書き込み可能であることを意味する。ｍ，ｔ，ｄ，およびＰは、以下のとおりである（各ケースにおいて、ｖは、Ｍ．ｘ．ｅであるとし、Ｄ．ｘ ＝ Ｄ１．ｘ ∨ Ｄ２．ｘであるとする）。

【００４９】

【表１】

- If ¬ R.x, m.x = t.x = d.x = undefined, where “undefined” is a pfn of a page not in physical memory.
- d.x = if D1.x then x else undefined
- ¬ D2.x ⇒ t.x = undefined
- P.(m.x).e = v (i.e., in P, m.x looks exactly like x does in M)
- d.x ≠ undefined ⇒ P.(d.x).e = if v.present then v [pfn ← t.(v.pfn)] else v
- t.x ≠ undefined ⇒ P.(t.x).e = if v.present then v [pfn ← m.(v.pfn), rw ← (v.rw ∧ (R.(v.pfn) ⇒ W.(v.pfn)) ∧ ¬ D.(v.pfn))] else v

【００５０】

言い換えると、ｘのディレクトリーバージョンは、ちょうどｘのメモリバージョンのよ

うに見えるが、ＰＦＮによりテーブルバージョンへ書き直される（ｒｅｄｉｒｅｃｔｅｄ）ＰＦＮをともなっている。ｘのテーブルバージョンは、ちょうどｘのメモリバージョンのように見えるが、複数のバージョンを読み取るために書き直されたＰＦＮをともなうとともに、読み出し可能であるが書き込み不可であるターゲットに対しクリアされた、あるいはＤにある読み出し－書き込みビットをともなっている。（もしくは、エントリーの表現が読み出し／書き込みビットを含んでいるかまたは読み出し専用ビットを含んでいるかに依存して、そのようなターゲットに対しセットされた他の読み出し専用ビットをともなっている）ディレクトリーに対し、ｘに関しｘのディレクトリーバージョンを保持する。しかし、テーブルのために、Ｄ２からｘを除去する（例えば、それをディスクにスワップする）コストを最小にするため、ｘに関しｘのリードバージョンを保持することに注意されたい。バージョンが、（例えば、ＡＴＣが許可したディレクトリーとテーブルのために）たまたま同じデータを保持する場合は常に、それらは同じ物理ページを共有する。そのため、ＡＴＣが拒絶もしくは変更する書き込みを行おうとしないソフトウェアオブジェクトに対し、シャドウページは作成される必要はない。

【００５１】

図６と７は、それぞれ、ｄ（ｘ）およびｔ（ｘ）を作成するための処理例を示している。

【００５２】

ここで、図６を参照すると、マップ（すなわち、上述のマップＭ）の一部であるページｘが存在すると仮定されている。そして、それは、ｘに基づいたページｄ（ｘ）を作成することが要求されている。最初に、ページｘが適切なポリシーの下で読み出し可能であるかどうか判断される（６０２）。もし、ｘが読み出し可能でない場合、ｄ（ｘ）は、未定義となり（６０６）、この処理は終了する。一方、ｘが読み出し可能である場合は、ｘがＤ１のメンバーであるかどうか（すなわち、ｘがページディレクトリーとして使用可能であるものとして指定されていたかどうか）判断される。もし、ｘがＤ１のメンバーでない場合、ｄ（ｘ）は、未定義である（６０６）。ｘがＤ１のメンバーである場合、ページｄ（ｘ）は、存在とマークされたエントリー中のＰＦＮフィールドが、それらのターゲットのテーブルバージョンを指すように変更されるのを除いて、ページｘと同じ内容を含むように作成される。この結果は、ｎ＝０と設定することによって達成することができる（６０７）、次いで、所与のｎの値に対し、ｘの第ｎエントリーが存在とマークされているかどうか判断する。もし、ｘの第ｎエントリーが存在とマークされていない場合、ｄ（ｘ）の第ｎエントリーは、ｘの第ｎエントリーに等しくなるように設定することができる（６１０）。ｘの第ｎエントリーが存在とマークされている場合、ｄ（ｘ）の第ｎエントリーは、ＰＦＮフィールドがテーブルバージョンを指すために変更される点を除いて、ｘの第ｎエントリーと等しくなるように設定される（６１２）。（すなわち、もし、Ｐ．ｎ．ｐｆｎが、ページＰのＰＦＮフィールドを参照する場合、および、ｖが上述の意味をもつ場合、ｄ（ｘ）．ｎ．ｐｆｎ＝ｔ（ｖ．ｐｆｎ）である）。ｄ（ｘ）の第ｎエントリーが設定された後、ｎがインクリメントされ（６１４）、この処理は、次のエントリーを設定するために６０８に戻る。

【００５３】

ここで、図７を参照すると、マップの一部であるページｘが存在し、かつ、ページｔ（ｘ）を作成することが要求されるということが再び仮定されている。はじめに、ｘが適切なポリシーの下で読み出し可能であるかどうか判断される（７０２）。もし、ｘが読み出し可能でない場合、ｔ（ｘ）は、未定義の値に設定される（７０６）。そしてこの処理は終了する。このポリシーの下で読み出し可能である場合、ｘがＤ２のメンバーであるかどうか判断される（７０４）。もし、ｘがＤ２のメンバーでない場合、ｔ（ｘ）は、未定義に設定される（７０６）。ｘがＤ２のメンバーである場合、存在とマークされたページのＰＦＮが、ターゲットページのメモリバージョンを指すように調節される点と、特定の読み出し／書き込みリンクが、それらを読み出し専用にするために調節される点を除いて、ページｔ（ｘ）は、エントリーの値がｘのエントリーの値に等しくなるように作成される

。ページｔ（ｘ）に対しこの内容を作成するために、最初に、カウンタｎが０に設定される（７０８）。次いで、ページｘの第ｎエントリーが存在にマークされているかどうか判断される。もし、このエントリーが存在とマークされていない場合、ｔ（ｘ）の第ｎエントリーは、ｘの第ｎエントリーに等しくなるように設定される（７１２）。第ｎページが存在とマークされている場合、ページｔ（ｘ）の第ｎエントリーは、このエントリーのＰＦＮフィールドが、ターゲットページのメモリバージョンを指すように設定される点を除いて、ｘの第ｎエントリーに等しくなるように設定される（７１４）（すなわち、もし、ｘの第ｎエントリーのターゲットページがＰＦＮ＝Ａをもつ場合、ｔ（ｘ）の第ｎエントリーのＰＦＮフィールドがｍ（Ａ）に等しくなるように設定される）。（上で述べたように、ｍ（Ａ）のＰＦＮは、しばしばＡのＰＦＮと等しくなる。）次に、第ｎエントリーのターゲットページが、適切なポリシーの下で読み出し可能だが書き込み不可であるページであるかどうか判断される（７１６）。もし、ターゲットページが読み出し可能で書き込み不可である場合、ｔ（ｘ）の第ｎエントリーは、読み出し専用としてマークされる（７２０）。そうでなければ、ｘの第ｎエントリーのターゲットページがＤ１またはＤ２のメンバーであるかどうか判断される（７１８）。もし、ｘの第ｎエントリーのターゲットページがＤ１またはＤ２のメンバーであるならば、ｔ（ｘ）の第ｎエントリーは、読み出し専用としてマークされる（７２０）。次いで、カウンターｎがインクリメントされる（７２２）。そして、ｔ（ｘ）の次のエントリーを作成するために、処理は７１４へ戻りループする。

【００５４】

（シャドウページの格納）

ページが、より効率的に表されるのを可能にする最適化があるが、各ページの３つのコピー（すなわち、ｄ（ｘ），ｔ（ｘ），およびｍ（ｘ））を格納することができる。第一に、もし、上述のアルゴリズムが、すでに格納されているバージョンと同一であるシャドウページを結果として作成する場合、シャドウページは、作成される必要は無い。したがって、ほとんどのページに対し、ページの１つのバージョンのみが格納される必要がある。どの場合においても、このようなページｘに対し、ｄ（ｘ），ｔ（ｘ），およびｍ（ｘ）のＰＦＮはすべて同じである。

【００５５】

第二に、ディレクトリーとは違ったすべてのページに対し、ページｘのオリジナルコピーがページのデータバージョンとして役立つことは好ましい。したがって、ディレクトリーではないページ（すなわち、フレーム番号が、Ｄ１のメンバーでないページ）に対し、ｍ（ｘ）のＰＦＮを、ｘのＰＦＮは等しく、ｘのディレクトリーバージョンおよびテーブルバージョンは他の所に格納される。しかしながら、ディレクトリーページの場合では、ページのオリジナルの位置がこのページのディレクトリーバージョンとして役立つことが好ましい（場合によっては、必要である）。Ｄ１組は、ディレクトリーとして仕えることが許されたページ（例えば、インテル（登録商標）ｘ８６プロセッサ上で、ＰＦＮをＣＲ３にロードすることができるページ）のＰＦＮに関して定義されているので、ディレクトリーバージョンを他のＰＦＮに移動させるのは現実的ではない。ＣＲ３は、シャドウページの存在を知らないかもしれないソフトウェアオブジェクトによってロードされることが必要である（例えば、ＣＲ３は、メモリアクセスがＡＴＣシステムによって制限されているオペレーティングシステムによってロードされるかもしれない）ので、ディレクトリーページは、ソフトウェアオブジェクトが、それらのページが配置されていると信じるＰＦＮに配置される必要があるかもしれない。

【００５６】

（ラージページに関してのシャドウページの使用）

上述のように、インテル（登録商標）ｘ８６プロセッサ（ならびに様々な他のプロセッサ）は、ラージページの使用をサポートする。その場合、どのページテーブルも、仮想アドレスを物理アドレスに変換することに関係していない。上述のシャドウイングメカニズムにラージページを機能させるために、たとえ、このシャドウページテーブルがどの実ペ

ージテーブルとも対応しないとしても、１つのシャドウページテーブルを１つのラージページのために作成することができる。したがって、ページディレクトリーｘが、ラージリンクを含む場合、そのページのディレクトリーバージョン（すなわち、ｄ（ｘ））を、ｘのラージリンクに対応するエントリーにスモールリンクを含むように作成することができる。このスモールリンクは、シャドウページテーブル（すなわち、ｔ（ｘ））を指す。そしてこのシャドウページテーブルは、ラージページを作り上げる個々のスモールページへのリンクを含む。上述のシャドウイングアルゴリズムの一部は、特定のページを読み出し専用としてマークすることを含むので、ラージページをスモールページに解体することは、全ラージページを読み出し専用ページとしてマークしなければならないことを回避する。ラージページの一部である個々のスモールページは、必要なら読み出し専用にマークすることができる。（読み出し専用としてラージページをマークすることの不利益は、そのようなページへの各書き込みリクエストが例外を生成し、その書き込みが該当する不変条件に違反することなしに行われることができるかどうか判断するために、より特権的なコンポーネント（例えば、ＡＴＣを実行するコンポーネント）によって評価されなければならないという点である。この方法で処理されるべきラージページへの各書き込みリクエストを要求することは、システムパフォーマンスをきわめて低下させる可能性がある。）

【００５７】

以上の例は単に説明のために示されており、本発明に関し限定を付すものであると解釈されてはならないことに留意されたい。本発明を様々な実施形態に関連して説明してきたが、ここで使用した用語は説明と例示に関する用語であり、限定するものではないことは理解されよう。さらに、本発明を特定の手段、素材、実施形態に関連して説明してきたが、本発明はここに開示する詳細に限定する意図はない。むしろ本発明は、特許請求の範囲内にあるような、機能的に同等のあらゆる構造、方法、使用方法に拡張される。本明細書の示唆する利点を理解する当業者は、これに様々な変更を加えることができる。こうした変更は、本発明の態様において本発明の範囲と精神を逸脱することなく実施することができる。

【図面の簡単な説明】

【００５８】

【図１】本発明の態様を実施できる一例のコンピューティング環境のブロック図である。

【図２】一例の仮想アドレスシステムのブロック図である。

【図３】アドレス変換マップにおける一例のエントリーのブロック図である。

【図４】アドレス変換制御を介してメモリアクセス制御のための一例の不変条件のブロック図である。

【図５】複数のバージョンに存在するページのブロック図である。ただし、ページの異なるバージョンが、このページが使用される状況に依存して使用される。

【図６】ページのディレクトリーバージョンを得るための一例の処理のフロー図である。

【図７】ページのテーブルバージョンを得るための一例の処理のフロー図である。

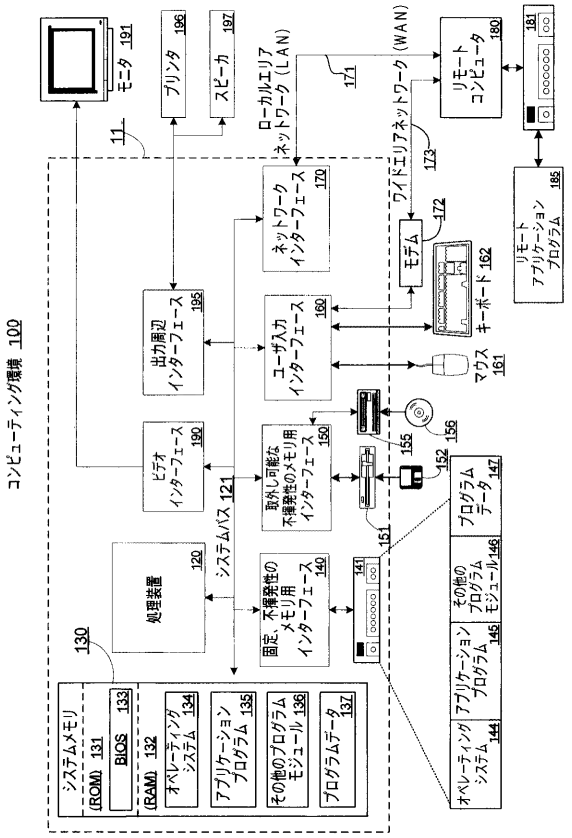
【符号の説明】

【００５９】

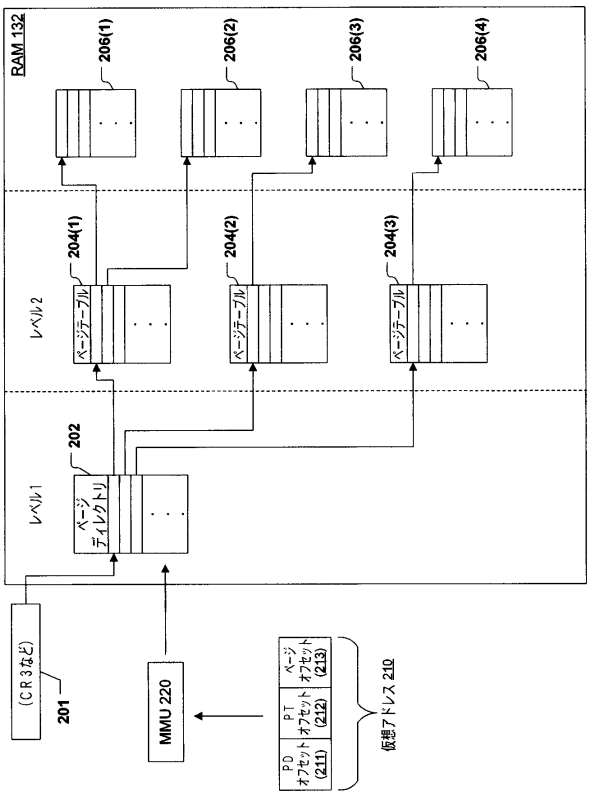
１００　コンピューティング環境
１１０　コンピュータ
１２０　処理装置
１２１　システムバス
１３０　システムメモリ
１３１　ＲＯＭ
１３２　ＲＡＭ
１３３　ＢＩＯＳ
１３４　オペレーティングシステム
１３５　アプリケーションプログラム
１３６　その他のプログラムモジュール

１３７　プログラムデータ
１４０　固定、不揮発性のメモリ用インターフェース
１４４　オペレーティングシステム
１４５　アプリケーションプログラム
１４６　その他のプログラムモジュール
１４７　プログラムデータ
１５０　取外し可能な不揮発性のメモリ用インターフェース
１６０　ユーザ入力インターフェース
１６１　マウス
１６２　キーボード
１７０　ネットワークインターフェース
１７１　ローカルエリアネットワーク（ＬＡＮ）
１７２　モデム
１７３　ワイドエリアネットワーク（ＷＡＮ）
１８０　リモートコンピュータ
１８５　リモートアプリケーションプログラム
１９０　ビデオインターフェース
１９１　モニタ
１９５　出力周辺インターフェース
１９６　プリンタ
１９７　スピーカ

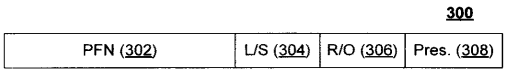
【図１】



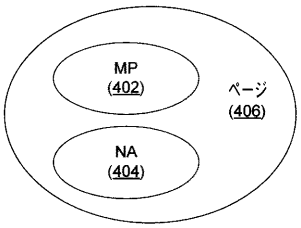
【図２】



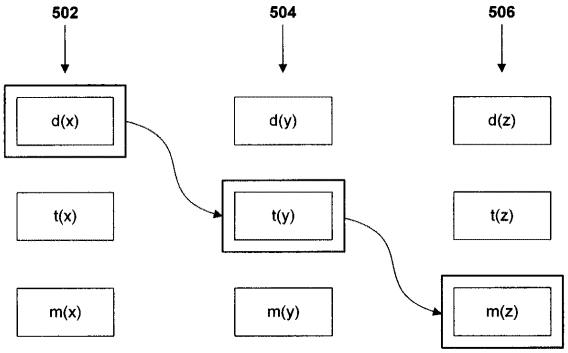
【図３】



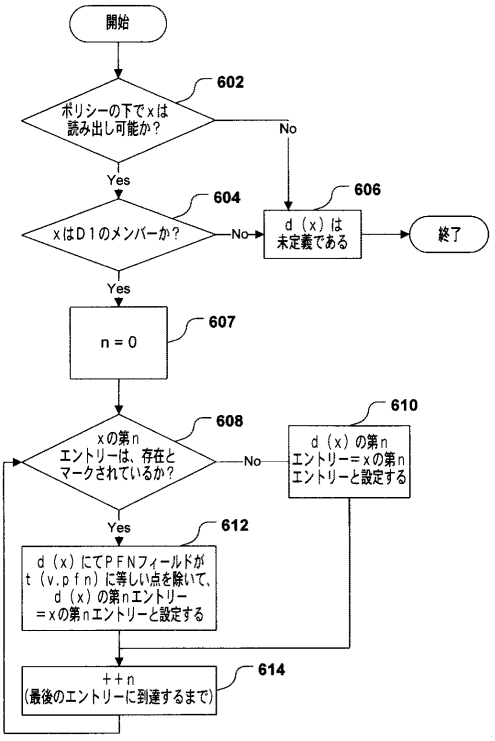
【図４】


MP
(402)
NA
(404)
ページ
(406)

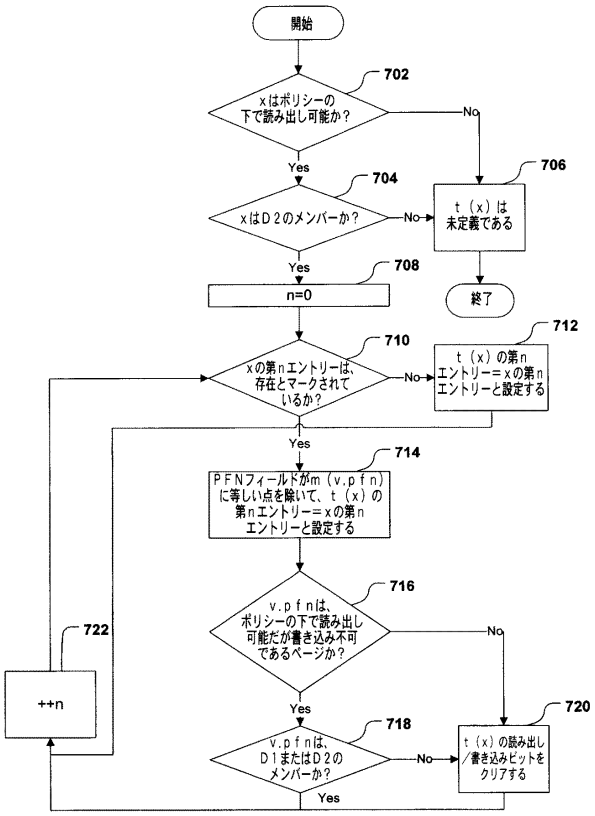
【図５】


502
504
506
d(x)
d(y)
d(z)
t(x)
t(y)
t(z)
m(x)
m(y)
m(z)

【図６】


開始
602
ポリシーの下でxは読み出し可能か?
No
Yes
604
xはD1のメンバーか?
No
606
d(x)は未定義である
終了
Yes
607
n = 0
608
xの第nエントリーは、存在とマークされているか?
No
610
d(x)の第nエントリー=xの第nエントリーと設定する
Yes
612
d(x)にてPFNフィールドがt(v.pfn)に等しい点を除いて、d(x)の第nエントリー=xの第nエントリーと設定する
++n
(最後のエントリーに到達するまで)
614

【図７】


開始
702
xはポリシーの下で読み出し可能か?
No
Yes
704
xはD2のメンバーか?
No
706
t(x)は未定義である
Yes
708
n=0
終了
710
xの第nエントリーは、存在とマークされているか?
No
712
t(x)の第nエントリー=xの第nエントリーと設定する
Yes
714
PFNフィールドがm(v.pfn)に等しい点を除いて、t(x)の第nエントリー=xの第nエントリーと設定する
716
v.pfnは、ポリシーの下で読み出し可能だが書き込み不可であるページか?
No
Yes
722
++n
718
v.pfnは、D1またはD2のメンバーか?
No
720
t(x)の読み出し/書き込みビットをクリアする
Yes

フロントページの続き

審査官　野田　佳邦

(56)参考文献　特開２００３－２５６２７８（ＪＰ，Ａ）
特開平１０－３１２３３８（ＪＰ，Ａ）
特開昭６２－０９９８４４（ＪＰ，Ａ）
米国特許第０６２３３６６８（ＵＳ，Ｂ１）
米国特許出願公開第２００３／０１７７４３５（ＵＳ，Ａ１）
米国特許出願公開第２００２／０１１６５９０（ＵＳ，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１２／０８－１２／１２
Ｇ０６Ｆ　１２／１６