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

特許第4882005号
(P4882005)

(45)発行日　平成24年2月22日(2012.2.22)　(24)登録日　平成23年12月9日(2011.12.9)

(51)Int.Cl.　　F I
G06F 9/46 (2006.01)　　G06F 9/46 ３５０
G06F 9/48 (2006.01)　　G06F 9/46 ３１１Ｚ

請求項の数 20　(全 21 頁)

(21)出願番号　特願2009-540365 (P2009-540365)
(86)(22)出願日　平成19年11月13日 (2007.11.13)
(65)公表番号　特表2010-512577 (P2010-512577A)
(43)公表日　平成22年4月22日 (2010.4.22)
(86)国際出願番号　PCT/US2007/084522
(87)国際公開番号　WO2008/070410
(87)国際公開日　平成20年6月12日 (2008.6.12)
審査請求日　平成22年10月15日 (2010.10.15)
(31)優先権主張番号　11/635,455
(32)優先日　平成18年12月6日 (2006.12.6)
(33)優先権主張国　米国(US)

早期審査対象出願

(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)復代理人　100115624
弁理士　濱中　淳宏
(74)復代理人　100162950
弁理士　久下　範子
(74)代理人　100077481
弁理士　谷　義一
(74)代理人　100088915
弁理士　阿部　和夫

最終頁に続く

(54)【発明の名称】仮想環境における最適化した割り込み送信

(57)【特許請求の範囲】
【請求項１】
仮想コンピューティングシステムに対する割り込みを処理する方法であって、前記仮想コンピューティングシステムは少なくとも１つのプロセッサー、仮想マシンモニター、およびゲストオペレーティングシステムと仮想割り込みコントローラーとを含む少なくとも１つの仮想マシンを備え、前記方法は
仮想割り込みコントローラーが、前記仮想マシンモニターから第１の割り込み要求を受信するステップと、
前記仮想割り込みコントローラーが、前記第１の割り込み要求に対応する割り込みサービスフラグを立て、前記第１の割り込み要求が、ゲストオペレーティングシステムにより自動ＥＯＩとしてプログラムされている場合、前記第１の割り込み要求をプロセッサーに送信すると、前記割り込みサービスフラグをクリアするステップと
を含むことを特徴とする方法。
【請求項２】
前記第１の割り込み要求を受信したプロセッサーが、前記第１の割り込み要求により規定される動作を実行するステップをさらに含み、前記割り込みサービスフラグをクリアするステップは、前記第１の割り込み要求が規定した動作が完全に実行される前に、前記第１の割り込み要求に対応する前記割り込みサービスフラグをクリアするステップを含むことを特徴とする請求項１に記載の方法。
【請求項３】

前記第１の割り込み要求を受信したプロセッサーが、前記第１の割り込み要求が規定した動作を実行するステップと、
前記第１の割り込み要求が規定した前記動作を実行する間、前記仮想割り込みコントローラーが、他の割り込み要求を受信するステップと
をさらに含むことを特徴とする請求項１に記載の方法。
【請求項４】
前記第１の割り込み要求が規定した動作を実行するステップは、前記仮想マシンモニターにより調停されることを特徴とする請求項３に記載の方法。
【請求項５】
前記仮想コンピューティングシステムは、第１のゲストオペレーティングシステムと、第２のゲストオペレーティングシステムとを備え、前記仮想マシンモニターは、前記第１のゲストオペレーティングシステムを含む第１のパーティションと、前記第２のゲストオペレーティングシステムを含む第２のパーティションを定義し、
前記第１の割り込み要求は、前記第１のパーティションから前記第２のパーティションへの第１のパーティション間メッセージに対する要求に対応することを特徴とする請求項１に記載の方法。
【請求項６】
各ゲストオペレーティングシステムが、仮想プロセッサーと関連する少なくとも１つのメッセージスロットを備え、
前記第１のパーティション間メッセージの処理が完了する前に第２のパーティション間メッセージが前記第１の割り込み要求の仮想プロセッサーに関連する前記メッセージスロットに対する待ち行列に入れられる場合、前記仮想マシンモニターが、前記第１のパーティション間メッセージに関連するメッセージ保留フラグをセットするステップ
をさらに含むことを特徴とする請求項５に記載の方法。
【請求項７】
前記パーティション間メッセージに関連するメッセージ保留フラグがセットされる場合、前記パーティション間メッセージを受信する仮想プロセッサーが、パーティション間メッセージの処理が完了した後にメッセージ終了命令を前記仮想割り込みコントローラーに送信するステップをさらに含むことを特徴とする請求項６に記載の方法。
【請求項８】
仮想コンピューティングシステムに対する割り込みを処理する方法であって、前記仮想コンピューティングシステムは仮想マシンモニター、および仮想割り込みコントローラーとゲストオペレーティングシステムとを含む少なくとも１つの仮想マシンを備え、前記方法は
前記仮想マシンモニターが、割り込みを前記ゲストオペレーティングシステムにリダイレクトするステップと、
前記仮想割り込みコントローラーが、前記リダイレクトされたゲストオペレーティングシステムから割り込み終了命令を受信するステップと、
前記仮想割り込みコントローラーが、前記割り込み終了命令は、サービス中としてフラグを立てた最高優先度の割り込みに対応しないことを判定するステップと、
前記仮想割り込みコントローラーが、前記受信した割り込み終了命令に対応する前記割り込みを特定する情報を記録するステップと、
サービス中としてフラグを立てた最高優先度の割り込みに対する割り込み終了命令を受信した後に、前記仮想割り込みコントローラーが、前記割り込み終了命令を処理するステップと
を含むことを特徴とする方法。
【請求項９】
サービス中の最高優先度の割り込みに対する割り込み終了命令の処理が完了した際に、前記仮想割り込みコントローラーが、前記記録した情報に対応する割り込みに対する割り込み終了命令を処理するステップをさらに含むことを特徴とする請求項８に記載の方法。

【請求項１０】
少なくとも１つのプロセッサーと、
仮想マシンモニターと、
ゲストオペレーティングシステムおよび仮想割り込みコントローラーを含む仮想マシンと
を備えたシステムであって、
前記仮想割り込みコントローラーは、前記仮想マシンモニターから割り込み要求を受信した後に、前記割り込み要求に対応する割り込みサービスフラグを立て、前記第１の割り込み要求が、ゲストオペレーティングシステムにより自動ＥＯＩとしてプログラムされている場合、前記第１の割り込み要求をプロセッサーに送信すると、前記割り込みサービスフラグをクリアすることを特徴とするシステム。
【請求項１１】
前記仮想割り込みコントローラーは、第１の割り込み要求が規定した動作が完全に実行される前に、前記割り込み要求に対応する割り込みサービスフラグをクリアすることを特徴とする請求項１０に記載のシステム。
【請求項１２】
前記仮想コンピューティングシステムは、第１のゲストオペレーティングシステムと、第２のゲストオペレーティングシステムとを備え、前記仮想マシンモニターは、前記第１のゲストオペレーティングシステムを含む第１のパーティションと、前記第２のゲストオペレーティングシステムを含む第２のパーティションを定義し、前記第１の割り込み要求は、前記第１のパーティションから前記第２のパーティションへの第１のパーティション間メッセージに対する要求に対応し、
前記仮想マシンモニターは、記第１のパーティションから前記第２のパーティションへの第１のパーティション間メッセージの処理が完了する前に第２のパーティション間メッセージが待ち行列に入れられる場合、前記第１のパーティション間メッセージにフラグを立て、前記第１のパーティション間メッセージにフラグが立てられている場合に限り、前記パーティション間メッセーを受信するプロセッサーが前記第１のパーティション間メッセージの処理完了時に前記仮想割り込みコントローラーに信号を送ることを特徴とする請求項１０に記載のシステム。
【請求項１３】
割り込みをゲストオペレーティングシステムにリダイレクトする仮想マシンモニターと、
ゲストオペレーティングシステムおよび仮想割り込みコントローラーを含む少なくとも１つの仮想マシンと
を備えたシステムであって、
前記仮想割り込みコントローラーは、前記リダイレクトされたゲストオペレーティングシステムから割り込み終了命令を受信し、前記割り込み終了命令は、サービス中としてフラグを立てた最高優先度の割り込みに対応しないことを判定し、前記受信した割り込み終了命令に対応する割り込みを特定する情報を記録して、サービス中のより高い優先度の割り込み処理完了後に前記受信した割り込み終了命令を処理可能とすることを特徴とするシステム。
【請求項１４】
コンピューターに、仮想コンピューティングシステムとして機能させるためのプログラムを備えたコンピューター読み取り可能な記録媒体であって、前記仮想コンピューティングシステムは、少なくとも１つのプロセッサー、仮想マシンモニター、およびゲストオペレーティングシステムと仮想割り込みコントローラーとを含む少なくとも１つの仮想マシンを備え、
前記仮想割り込みコントローラーは、前記仮想マシンモニターから割り込みサービス要求を受信した後に、前記割り込みサービス要求に対応する割り込みサービスフラグを立て、前記第１の割り込み要求が、ゲストオペレーティングシステムにより自動ＥＯＩとして

プログラムされている場合、前記割り込みサービス要求をプロセッサーに送信すると、前記割り込みサービスフラグをクリアするよう動作することを特徴とするコンピューター読み取り可能な記録媒体。

【請求項１５】

前記コンピューターに、割り込みサービス要求が規定した動作を実行させるためのプログラムをさらに備え、

前記仮想割り込みコントローラーは、前記コンピューターが前記割り込みサービス要求が規定した前記動作を実行する間、他の割り込みサービス要求を受信するようさらに動作することを特徴とする請求項１４に記載のコンピューター読み取り可能な記録媒体。

【請求項１６】

前記仮想マシンモニターは、第１のプロセッサー間メッセージの処理が完了する前に、第２のプロセッサー間メッセージが前記第１のプロセッサー間メッセージを受信するプロセッサーに対する待ち行列に入れられる場合、前記第１のプロセッサー間メッセージに関連するメッセージ保留フラグをセットするよう動作することを特徴とする請求項１４に記載のコンピューター読み取り可能な記録媒体。

【請求項１７】

第１のプロセッサー間メッセージに関連する前記メッセージ保留フラグを、前記第１のプロセッサー間メッセージのヘッダ内のビットとして具現化したことを特徴とする請求項１６に記載のコンピューター読み取り可能な記録媒体。

【請求項１８】

前記プロセッサー間メッセージを受信するプロセッサーは、プロセッサー間メッセージに関連するメッセージ保留フラグがセットされている場合、前記プロセッサー間メッセージの処理が完了した後に仮想割り込みコントローラーに信号を送るよう動作することを特徴とする請求項１６に記載のコンピューター読み取り可能な記録媒体。

【請求項１９】

コンピューターに、仮想マシンモニター、およびゲストオペレーティングシステムと仮想割り込みコントローラーとを含む少なくとも１つの仮想マシンを備えた仮想コンピューティングシステムとして機能させるためのプログラムを備えたコンピューター読み取り可能な記録媒体であって、

前記仮想マシンモニターは、割り込みを前記ゲストオペレーティングシステムにリダイレクトするよう動作し、

前記仮想割り込みコントローラーは、前記リダイレクトされたゲストオペレーティングシステムから割り込み終了命令を受信し、前記割り込み終了命令を受信したときに処理中である最高優先度の割り込みに前記割り込み終了命令が対応しない場合、後に処理する前記割り込み終了命令を待ち行列に入れるよう動作することを特徴とするコンピューター読み取り可能な記録媒体。

【請求項２０】

前記仮想割り込みコントローラーは、待ち行列に入れた割り込み終了命令に対応する割り込みより優先度が高い割り込みに対応する割り込み終了命令の処理後、前記待ち行列に入れた割り込み終了命令を処理するようさらに動作することを特徴とする請求頃１９に記載のコンピューター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、仮想環境における最適化した割り込み送信に関する。

【背景技術】

【０００２】

バーチャルマシン（ＶＭ）は、エミュレートされたマシンまたはシステムを提供するためにコンピューターデバイスなど（例えば、ホスト）で動作するソフトウェア構成またはそれと同種のものである。一般に、必ずではないが、ＶＭはアプリケーションなどであり

、ホスト上で使用して利用アプリケーションなどをインスタンス化し、同時に上記利用アプリケーションを上記ホスト装置から、または上記ホスト上の他のアプリケーションから隔離することができる。１つの一般的な状況では、ホストは配備されている複数のＶＭを収容することができ、それぞれのＶＭはホストから利用可能なリソースを用いて何らかの所定の機能を実行する。

【０００３】

特に、コンピューターデバイス上にホストされるそれぞれのＶＭは、仮想的な形ではあるが実質的にコンピューティングマシンであり、従ってその利用アプリケーションおよび外部の両方に対して自身をコンピューティングマシンの様に見せる。例えば、ＶＭおよび／またはその利用アプリケーションは、ＶＭにハードウェアリソースが実際にはなくても、ＶＭのハードウェアリソースにハードウェア要求を発行することができ、実際にそうしている。理解できるように、上記ハードウェア要求は遮断されるか、またはホストにリダイレクトされ、上記ホストがそのハードウェアリソースに基づいて上記ハードウェア要求を処理する。このことを、要求したＶＭおよび／またはその利用アプリケーションは、一般に全く認識しない。

【０００４】

一般に、必ずではないが、ホストはそのそれぞれのＶＭを別々のパーティション、アドレス空間、処理領域、などに展開する。上記ホストは、仮想マシンモニター（「ＶＭＭ」）などを有する仮想化層を含むことができる。ＶＭＭは監視アプリケーションまたは「ハイパーバイザ」として動作し、仮想化層はホストのそれぞれのＶＭの監視態様を監督および／または管理する。さらにＶＭＭは、それぞれのＶＭと外部との間を接続しうるものとして動作する。ＶＭＭは、自分のアドレス空間で実行される独立したアプリケーションとすることができ、または、直接的に、もしくはデバイスドライバなどのある種のオペレーティングシステム拡張として、ＶＭをホストオペレーティングシステムと、より密接に統合することができる。特にホストのＶＭＭは、ホストのそれぞれのＶＭおよび／またはその利用アプリケーションからのハードウェア要求を遮断またはリダイレクトすることができ、少なくともその要求処理を支援することができるが、先と同様にこのことを、要求したＶＭおよび／またはその利用アプリケーションは全く認識しない。

【０００５】

多くのコンピューティングシステムは複数のプロセッサーを含む。マルチプロセッサーの仮想マシン環境内のプロセッサーは、ゲストモードまたはＶＭＭモードで動作することができる。ゲストモードで実行されるとき、プロセッサーは仮想マシンの定義を使用して、仮想マシンのゲストオペレーティングシステムとアプリケーションとを管理し、ＶＭＭからの干渉を受けずに引数を変換し、システムリソースを管理する。時々、ゲストオペレーティングシステムまたはアプリケーションが、ＶＭＭによる管理が必須のシステムリソースを必要とする場合がある。例えば、ＶＭＭがエラー処理、システム障害、または割り込み処理を要求される可能性がある。これらの状況では、プロセッサーはＶＭＭモードで動作する。

【０００６】

近代の処理システムでは割り込みがサポートされ、それによりプロセッサーに外部のイベントを通知することができる。例えば、ユーザーがキーボードのキーを押下し、またはネットワークパケットが電線から到着すると、対応する割り込みが生成されてプロセッサーに送信される。一般に、割り込みによりプロセッサーは現在実行中のものを停止し、現在の実行位置を記録する。それによりプロセッサーは割り込み処理後に実行を再開し、特定の割り込みサービスルーチンを実行することができる。

【０００７】

コンピューティングシステムは、システム内での割り込みの流れを指示および調停する１つまたは複数の割り込みコントローラーを含むことができる。割り込みコントローラーのロジックを別々のハードウェアコンポーネントに具現化すること、プロセッサーに統合すること、または仮想化することができる。割り込みコントローラーは、特に、マルチプ

ロセッサー環境において割り込み優先度を決定し、割り込みを適切なプロセッサーへ送信する責任を有する。仮想環境では、プロセッサーおよび割り込みコントローラーを仮想化することができる。これは一般に、仮想マシンモニターのようなソフトウェアと、ハードウェアが提供する仮想化支援機構（virtualization assist）との組合せにより実現される。

【０００８】

一般に、割り込みを処理した後、割り込み終了（ＥＯＩ）命令を介して割り込みコントローラーに通知する。ＥＯＩ命令は、前の割り込みの処理中に送信が保留された可能性がある他の割り込みを、この時点で送信できることを割り込みコントローラーに伝える。ＥＯＩ命令は、一般にレジスタからの読み取りまたはレジスタへの書き込みなど、Ｉ／Ｏポートまたはメモリーマップド Ｉ／Ｏアクセスを通して割り込みコントローラーに送信される。物理割り込みコントローラーに関しては、ＥＯＩ命令の処理により何十何百ものサイクルが消費される可能性がある。仮想割り込みコントローラーに関しては、ＥＯＩ命令の処理により何千ものサイクルが消費される可能性がある。

【０００９】

仮想マシンモニターの中には、割り込みをパーティション間メッセージングの基礎として使用するものがある。１つのパーティション内で実行されるソフトウェアが同一物理マシン上の第２のパーティション内で実行されるソフトウェアと通信する必要がある場合、パーティション間メッセージングを用いて行うことができる。或るプロセッサーがメッセージを送信すると、仮想マシンモニターはメッセージの受信者であるプロセッサーに割り込みを送信し、受信プロセッサーの割り込みサービスルーチンにメッセージを処理させ、その内容に応答させることができる。

【発明の概要】

【００１０】

本概要は、以下の発明の詳細な説明でさらに説明されている選択した概念を簡略化した形式で紹介するために提供するものである。本概要は、請求項に記載されている主題事項の重要な特徴または本質的な特徴を特定することを目的とするものでも、請求項に記載されている主題事項の範囲を決定に際して、補助として使用することを目的とするものでもない。

【００１１】

本明細書では、マルチプロセッサーの仮想環境において割り込みを効率的に処理する機構を説明する。いくつかの実施形態では、ゲストオペレーティングシステムは或る割り込み源を「自動割込み終了」（自動ＥＯＩ）としてプログラムすることができる。自動ＥＯＩを処理する際、仮想割り込みコントローラーは明示的な割り込み終了（ＥＯＩ）命令を待たずに、送信された割り込みに対応する割り込みサービスレジスタ内のビットをクリアする。自動ＥＯＩ割り込みにより、他の割り込みの送信をブロックすることはできない。

【００１２】

割り込みを使用してパーティション間通信を実施することができる。ゲストオペレーティングシステムがパーティション間メッセージに関連する割り込みを受信すると、ゲストオペレーティングシステムの割り込みサービスルーチンが指定のメッセージスロットからメッセージを読み取り、そのメッセージタイプとペイロードとに基づいて動作を実行する。以下で詳述するように、ゲストオペレーティングシステムは、別のメッセージが同一メッセージスロットに対する待ち行列内にある場合に限り、明示的なメッセージ終了（「ＥＯＭ」）命令をメッセージ処理完了時に送信することで、パーティション間メッセージの処理により被るオーバヘッドを幾らか排除することができる。ＥＯＭ命令の計算コストは、大まかにＥＯＩの計算コストと等しいが、ＥＯＭが送信されるのは、追加のメッセージが待ち行列内にあるという稀なケースに限られる。これにより、パーティション間通信に対する割り込み処理の平均的なコストを大幅に削減することができる。

【００１３】

割り込みは様々な優先度を有することができる。一般に、高優先度の割り込みは低優先

度の割り込み処理に割り込むことができるが、逆は不可能である。仮想環境では、サービス中の最高優先度の割り込みではない割り込みに対して、ゲストオペレーティングシステムがＥＯＩ命令を発行することが可能である。本明細書では、仮想割り込みコントローラーが、着信中のＥＯＩ命令がサービス中の最高優先度の割り込みに対応するかどうかをチェックする機構を説明する。着信中のＥＯＩ命令がサービス中の最高優先度の割り込みに対応しない場合、仮想割り込みコントローラーは、着信中のＥＯＩ命令に対する割り込みベクトルを、後にＥＯＩの発行が必要な割り込みの集合に加える。着信中のＥＯＩ命令がサービス中の最高優先度の割り込みに対応する場合、仮想割り込みコントローラーは対応する割り込みに対するＥＯＩ命令を処理するだけでなく、後にＥＯＩを発行するために前もってマークした全ての他の割り込みに対するＥＯＩ命令も処理する。

【図面の簡単な説明】

【００１４】

【図１】コンピューターシステムにおける仮想動作環境に対するハードウェアおよびソフトウェアアーキテクチャの論理階層を表すブロック図である。

【図２】仮想化がホストオペレーティングシステムにより（直接またはハイパーバイザを介して）行われる仮想コンピューティングシステムを表すブロック図である。

【図３】ホストオペレーティングシステムと並行して実行される仮想マシンモニターにより仮想化が行われる、代替的な仮想コンピューティングシステムを表すブロック図である。

【図４】ホストオペレーティングシステムとは独立に動作するバーチャライザーにより仮想化が行われる、別の代替的な仮想コンピューティングシステムを表すブロック図である。

【図５】割り込みコントローラーを有するコンピューティングシステムの一部を示すブロック図である。

【図６】割り込み要求の処理方法を示すフローチャートである。

【図７】割り込み優先度の例に対するタイムラインを示す図である。

【図８】本明細書の教示に従って自動ＥＯＩを用いた割り込み要求の処理方法を示すフローチャートである。

【図９】本明細書の教示に従ってプロセッサー間メッセージの処理方法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】

【００１５】

或る特定の詳細を、以下の記述および図面で説明して、本発明の様々な実施形態の徹底的な理解を与える。コンピューティング技術およびソフトウェア技術に関連することの多い或る公知な詳細は、様々な実施形態を不必要に不明瞭とすることを回避するため、以下の開示では説明していない。さらに、当業者は、後述する１つまたは複数の詳細がなくとも他の実施形態を実践できることを理解するであろう。最後に、以下の開示におけるステップおよび順序を参照して様々な方法を説明するが、そのような説明は本発明の実施形態を明確に実施するためであり、そのステップおよびステップの順序が本発明を実践するための要件であると捉えるべきではない。

【００１６】

本明細書で説明されている様々な技術を、ハードウェアもしくはソフトウェア、または必要に応じてその両方の組み合わせと関連させて実施できることは理解されるべきである。従って、方法および装置、またはその或る態様もしくは一部は、有形媒体で具現化したプログラムコード（例えば、命令）の形態を取ることができ、当該有形媒体には、フロッピー（登録商標）ディスク、ＣＤ－ＲＯＭ、ハードドライブ、または任意の他の機械読取可能記憶媒体などがある。プログラムコードをコンピューターなどのマシンにロードして実行すると、そのマシンは本発明を実践する装置となる。プログラム可能なコンピューター上でプログラムコードを実行する場合、コンピューターデバイスは、一般にプロセッサー、プロセッサーにより読み取り可能な記憶媒体（揮発性および不揮発性のメモリーおよ

び／または記憶要素を含む）を含み、少なくとも１つの入力装置、および少なくとも１つの出力装置を含む。１つまたは複数のプログラムは、例えばＡＰＩ、再利用可能コントロールなどを使用して、本発明に関して説明されているプロセスを実施または利用することができる。コンピューターシステムと通信するには、上記プログラムを、高レベルの手続き型言語またはオブジェクト指向プログラミング言語で実施することが望ましい。ただし、必要に応じてプログラムをアセンブリ言語または機械語で実施することができる。いずれにせよ、言語はコンパイル言語またはインタープリタ型言語に変換される言語であってもよく、ハードウェア装置と組み合わされてもよい。

【００１７】

例示的な実施形態は、１つまたは複数のスタンドアロンコンピューターシステムが存在する状況での本発明の態様の利用に関することができるが、本発明はそのようには限定されず、ネットワークまたは分散コンピューティング環境などの任意のコンピューティング環境と関連して実施することができる。さらに、本発明の態様を複数の処理チップまたは装置内で、またはそれらに渡って実施することができ、同様に複数の装置に渡って記憶装置に影響を及ぼすことができる。上記装置には、パーソナルコンピューター、ネットワークサーバー、ハンドヘルド装置、スーパーコンピューター、または自動車および航空機などの他のシステムに統合されるコンピューターを含めることができる。

【００１８】

概要

仮想マシン環境内で割り込みを効率的に処理する様々な方法およびシステムを説明する。割り込みは、現代のコンピューティングシステムにおいて、一例として、プロセッサーに外部イベントを通知すること、マルチプロセッサーシステムのプロセッサー間通信を促進することを含む、様々な目的で使用されている。一般に、割り込みは、通常の処理に割り込んで、一時的に制御フローを割り込みサービスルーチン（「ＩＳＲ」）に回す。コンピューティングシステムの様々な活動が割り込みを引き起こす可能性がある。いくつかの例として、キーボードのキーの押下、ネットワークパケットの受信、およびディスクの読み書きがある。プロセッサー間割り込み（「ＩＰＩ」）は一種の割り込みであり、それによりマルチプロセッサー環境において或るプロセッサーは別のプロセッサーに割り込むことができる。ＩＰＩをプロセッサー間メッセージングの基礎として使用することができる。

【００１９】

コンピューティングシステムは、一般にシステム内の割り込みの流れを指示および調停する１つまたは複数の割り込みコントローラーを含む。割り込みコントローラーは着信した割り込みの優先付け、それらの割り込みをマルチプロセッサー内の適切なプロセッサーに割り当てる責任を有する。割り込みコントローラーをハードウェアで実現することができ、従って割り込みコントローラーは別々のコンポーネントとすることができ、またはプロセッサーに統合することができる。割り込みコントローラーを仮想化してもよい。割り込みコントローラーの仮想化は、一般にソフトウェアと、ハードウェアが与える仮想化支援機構とを組み合わせて実現される。ソフトウェアは、物理割り込みコントローラーと同一の基本機能を行う仮想マシンモニターの一部であってもよい。

【００２０】

一般に、それぞれの割り込み源は、指定の割り込み優先度を有する。一実施例として、これらの優先度に０から２５５の番号を付与することができる。２５５は最高優先度であり、０は最低優先度である。高優先度の割り込みは、低優先度の割り込みを処理する割り込みサービスルーチンに割り込むことができるが、低優先度の割り込みは高優先度の割り込みに割り込むことはできない。割り込みサービスルーチンが実行を完了すると、それを実行していたプロセッサーは一般にＥＯＩ命令を発行して、割り込みコントローラーに割り込みの処理が完了したこと、および保留された低優先度の割り込みをこの時点で送信できることを知らせる。

【００２１】

仮想割り込みコントローラーでは、ＥＯＩ命令は、物理割り込みコントローラーがＥＯＩ命令に応答して行うものと同一の動作を行うソフトウェアで実施される。これには一般に、ＥＯＩポートまたはレジスタへのアクセスを遮断してソフトウェアハンドラを呼び出すことが含まれる。遮断およびソフトウェアハンドラを組み合わせると、一般にＥＯＩ命令の処理に何千何万ものサイクルが必要となり、仮想環境における割り込みサービスルーチンの動作に大量のオーバヘッドが加わる。

【００２２】

本明細書で説明する方法およびシステムにより、割り込みを効率的に処理する機構が実現される。多くの場合において、ＥＯＩ命令を省略し、割り込み送信に関する仮想化オーバヘッドを大幅に削減することができる。プロセッサー間メッセージングに用いるＩＰＩの場合、丁度処理したメッセージを含むスロットに対して第２のメッセージが既に待ち行列内にあるときに限り、メッセージＥＯＩの終了を送信する必要がある。いくつかの場合、物理割り込みに対して選択的にＥＯＩを送信することができるが、これはその物理割り込みがサービス中の最高優先度の割り込みであるかどうかに無関係である。

【００２３】

一般的な仮想化

オペレーティングシステムおよびプロセッサーの命令セットにおける多様性のため、ソフトウェアの相互運用性が低下する可能性がある。高レベル言語およびオペレーティングシステム双方におけるメモリーおよびＩ／Ｏ抽象化により、ハードウェアリソースへの依存性を幾分削減することができるが、幾分かの依存性は残ることになる。多くのオペレーティングシステムが特定のシステムアーキテクチャに対して開発され、ハードウェアリソースを直接管理するように設計されている。これにより、利用可能なソフトウェアおよびオペレーティングシステムに関してコンピューターシステムの柔軟性が限定される可能性があり、特にシステムが複数ユーザーにより共有されるとき、セキュリティおよび障害隔離に悪影響を及ぼす可能性がある。

【００２４】

仮想化により、セキュリティおよび信頼性を高めつつ柔軟性を向上させる機構が実現される。プロセッサー、メモリー、およびＩ／Ｏ装置は、仮想化できるサブシステムの例である。サブシステムを仮想化すると、仮想インターフェースと、その仮想インターフェースを通して利用可能な仮想リソースとが、仮想化を実施した実システムのインターフェースとリソースとにマッピングされる。仮想化を、サブシステムだけでなくマシン全体に適用することができる。仮想マシンのアーキテクチャは、実マシン上のソフトウェア層で実施される。

【００２５】

概念的観点からは、コンピューターシステムは、一般に基本的なハードウェア層で実行される１つまたは複数のソフトウェア層を含む。この階層化は抽象化のために行われている。所与のソフトウェア層に対してインターフェースを定義することで、その層をそれより上の他の層と別々に実施することができる。良く設計されたコンピューターシステムでは、それぞれの層は直下の層に関してのみ知っている（および、直下の層にのみ依存する）。これにより層または「スタック」（複数の隣接層）を、上記層またはスタックより上の層に悪影響を及ぼすことなく置換することができる。例えば、ソフトウェアアプリケーション（上層）は、一般に低レベルのオペレーティングシステム（低層）に依存して何らかの形の永久記憶装置にファイルを書き込むが、これらのアプリケーションはフロッピー（登録商標）ディスク、ハードドライブ、またはネットワークフォルダへデータを書き込むことの違いを理解する必要はない。この低層を、ファイル書き込み用の新しいオペレーティングシステムのコンポーネントで置換しても、上層のソフトウェアアプリケーションの動作は、影響を受けないままである。

【００２６】

階層化したソフトウェアの柔軟性により、ＶＭ（virtual machine）は、実際は別のソフトウェア層にある仮想ハードウェア層を表現することができる。このように、ＶＭはそ

れより上のソフトウェア層に対して、自分がプライベートなコンピューターシステム上で実行されているかのように錯覚させることができ、従って、ＶＭにより複数の「ゲストシステム」を単一の「ホストシステム」上で並列に実行することができる。

【００２７】

図１は、コンピューターシステム内の仮想環境に対するハードウェアおよびソフトウェアアーキテクチャの論理階層を表す図である。図１で、仮想化プログラム１１０は、物理ハードウェアアーキテクチャ１１２上で直接的または間接的に実行される。仮想化プログラム１１０は、（ａ）ホストオペレーティングシステムと並行に実行される仮想マシンモニターとすることもでき、または（ｂ）ハイパーバイザコンポーネントを有するホストオペレーティングシステムとすることができる。この場合、ハイパーバイザコンポーネントが仮想化を行う。仮想マシンモニターという用語は、様々な種類の仮想化プログラムのうち何れかに対する一般的な用語として用いる。仮想化プログラム１１０は、（このコンポーネントがパーティションまたは「仮想マシン」である事実を表すため点線で示してある）ゲストハードウェアアーキテクチャ１０８を仮想化する。即ち、ハードウェアは実際には存在しないが、その代わりに仮想化プログラム１１０により仮想化される。ゲストオペレーティングシステム１０６を、ゲストハードウェアアーキテクチャ１０８上で実行し、ソフトウェアアプリケーション１０４を、ゲストオペレーティングシステム１０６上で実行することができる。図１の仮想化した動作環境では、ソフトウェアアプリケーション１０４が、ホストオペレーティングシステムおよびハードウェアアーキテクチャ１１２と一般に非互換であるオペレーティングシステム上で実行するように設計されていても、ソフトウェアアプリケーション１０４をコンピューターシステム１０２内で実行することができる。

【００２８】

次に、図２は、物理コンピューターハードウェア２０２の直上で実行されるホストオペレーティングシステム（ホストＯＳ）のソフトウェア層２０４を含む仮想コンピューティングシステムを示す。ここで、ホストＯＳ　２０４は、オペレーティングシステムＡ　２１２およびオペレーティングシステムＢ　２１４によりそれぞれ使用されるパーティションＡ　２０８およびパーティションＢ　２１０にインターフェースを公開することで、物理コンピューターハードウェア２０２のリソースへのアクセスを提供する。これによりホストＯＳ　２０４は、その上で実行されるオペレーティングシステム層２１２および２１４に注目されずに済む。繰り返しになるが、仮想化を行うため、ホストＯＳ　２０４はネイティブな仮想化機能を持つ特殊設計のオペレーティングシステムとすることができ、あるいは仮想化を行うための組み込みのハイパーバイザコンポーネントを持つ標準的なオペレーティングシステムとすることができる（図示せず）。

【００２９】

再度図２を参照する。ホストＯＳ　２０４の上には、２つのパーティション、パーティションＡ　２０８およびパーティションＢ　２１０がある。パーティションＡ　２０８は、例えば仮想のインテル３８６プロセッサーとすることができ、パーティションＢ　２１０は、例えばモトローラ６８０Ｘ０系プロセッサーの１つを仮想化したものとすることができる。それぞれのパーティション２０８およびパーティション２１０内には、それぞれゲストオペレーティングシステム（ゲストＯＳ）Ａ　２１２およびゲストＯＳ　Ｂ　２１４がある。ゲストＯＳ　Ａ　２１２上では２つのアプリケーション、アプリケーションＡ１　２１６およびアプリケーションＡ２　２１８が実行され、ゲストＯＳ　Ｂ　２１４上ではアプリケーションＢ１　２２０が実行されている。

【００３０】

図２に関して、（点線で示す）パーティションＡ　２０８とパーティションＢ　２１４とは、ソフトウェア構造としてのみ存在する仮想コンピューターハードウェア表現であることに留意することが重要である。これらは、特殊な仮想化ソフトウェアを実行することで可能となるものである。当該ソフトウェアは、パーティションＡ　２０８およびパーティションＢ　２１０をそれぞれゲストＯＳ　Ａ　２１２およびゲストＯＳ　Ｂ　２１４に

提示するだけでなく、ゲストＯＳ　Ａ　２１２およびゲストＯＳ　Ｂ　２１４が実物理コンピューターハードウェア２０２と間接的に相互作用するために必要な、全てのソフトウェアステップも実行する。物理コンピューターハードウェア２０２は、単一プロセッサー環境の場合は単一のＣＰＵ（ｃｅｎｔｒａｌ　ｐｒｏｃｅｓｓｉｎｇ　ｕｎｉｔ）２２２を、マルチプロセッサー環境の場合は複数のＣＰＵ２２２、２２４、２２６を含むことができる。

【００３１】

図３は、ホストオペレーティングシステム３０６と並行に実行される仮想マシンマネージャ３０４により仮想化が行われる、代替的な仮想コンピューティングシステムを示す。或る場合、仮想マシンマネージャ３０４は、ホストオペレーティングシステム３０６上で実行され、ホストオペレーティングシステム３０６を通してのみコンピューターハードウェア３０２と相互作用するアプリケーションとすることができる。図３に示すような他の場合、仮想マシンマネージャ３０４は、代わりに部分的に独立したソフトウェアシステムを含むことができる。当該ソフトウェアシステムは、いくつかのレベルではホストオペレーティングシステム３０６を介してコンピューターハードウェア３０２と間接的に相互作用するが、他のレベルでは、仮想マシンマネージャ３０４は（ホストオペレーティングシステムがコンピューターハードウェアと直接相互作用するのと同様に）コンピューターハードウェア３０２と直接相互作用する。さらに他の場合、仮想マシンマネージャ３０４は、完全に独立したソフトウェアシステムを含むことができる。当該ソフトウェアシステムは、ホストオペレーティングシステム３０６を利用せずに、（ホストオペレーティングシステムがコンピューターハードウェアと直接相互作用するのと同様に）全てのレベルでコンピューターハードウェア３０２と直接相互作用する（しかしながら、コンピューターハードウェア３０２の使用を調整し、衝突を回避するなどのため、ホストオペレーティングシステム３０６とも相互作用する）。

【００３２】

図３に示す実施例では、２つのパーティション、パーティションＡ　３０８およびパーティションＢ　３１０は、概念的に仮想マシンマネージャ３０４の上にある。それぞれのパーティション３０８およびパーティション３１０内には、それぞれゲストオペレーティングシステム（ゲストＯＳ）Ａ　３１２およびゲストＯＳ　Ｂ　３１４がある。ゲストＯＳ　Ａ　３１２上では２つのアプリケーション、アプリケーションＡ１　３１６およびアプリケーションＡ２　３１８が実行され、ゲストＯＳ　Ｂ　３１４上ではアプリケーションＢ１　３２０が実行されている。物理コンピューターハードウェア３０２は、単一プロセッサー環境の場合は単一のＣＰＵ（ｃｅｎｔｒａｌ　ｐｒｏｃｅｓｓｉｎｇ　ｕｎｉｔ）３２２を、マルチプロセッサー環境の場合は複数のＣＰＵ３２２、３２４、３２６を含むことができる。

【００３３】

図４は、ハイパーバイザ４０４により仮想化が行われる別の代替的な仮想コンピューティングシステムを示す。ハイパーバイザ４０４は、ホストオペレーティングシステムを用いずにコンピューターハードウェア４０２と直接相互作用できる独立したソフトウェアシステムを含む。物理コンピューターハードウェア４０２は、単一プロセッサー環境の場合は単一のＣＰＵ（ｃｅｎｔｒａｌ　ｐｒｏｃｅｓｓｉｎｇ　ｕｎｉｔ）４２２を、マルチプロセッサー環境の場合は複数のＣＰＵ４２２、４２４、４２６を含むことができる。

【００３４】

図４に示す実施例では、２つのパーティション、パーティションＡ　４０８およびパーティションＢ　４１０は、概念的に仮想マシンマネージャ４０４の上にある。それぞれのパーティション４０８およびパーティション４１０内には、それぞれゲストオペレーティングシステム（ゲストＯＳ）Ａ　４１２およびゲストＯＳ　Ｂ　４１４がある。ゲストＯＳ　Ａ　４１２上では２つのアプリケーション、アプリケーションＡ１　４１６およびアプリケーションＡ２　４１８が実行され、ゲストＯＳ　Ｂ　４１４上ではアプリケーションＢ１　４２０が実行されている。ゲストＯＳ　Ａ　４１２はホストＯＳのサービスを提

供する。物理コンピューターハードウェア４０２は、単一プロセッサー環境の場合は単一のＣＰＵ（ｃｅｎｔｒａｌ　ｐｒｏｃｅｓｓｉｎｇ　ｕｎｉｔ）４２２を、マルチプロセッサー環境の場合は複数のＣＰＵ４２２、４２４、４２６を含むことができる。

【００３５】

上述のパーティションを実施するこれらの変形の全ては例示的な実施例に過ぎず、本開示の発明を特定の仮想化の態様に限定するものと解釈すべきではない。

【００３６】

一般的な割り込み処理

図５は、割り込みコントローラーを有するマルチプロセッサーコンピューティングシステムの一部分の実施例を示すブロック図である。任意の数の装置５０２、５０４、５０６、５０８が、割り込み要求源の役割を果たすことができる。装置５０２、５０４、５０６、５０８は、例えばキーボード、ディスクドライブ、ネットワークカードなどの物理装置とすることができ、または仮想装置とすることができる。割り込み要求を、プロセッサー５１０、５１２、５１４の何れかにより生成してもよい。

【００３７】

割り込みコントローラー５１６は、割り込み要求の処理を調停および指示する。割り込みコントローラー５１６は、プログラム可能な割り込みコントローラー（「ＰＩＣ」）または高度なプログラム可能な割り込みコントローラー（「ＡＰＩＣ」）などの物理装置とすることができる。あるいは、割り込みコントローラー５１６を仮想化してもよく、この場合、その機能はＶＭＭ内のソフトウェアハンドラなどのソフトウェアにより実行される。

【００３８】

大部分の割り込みコントローラーは、要求されたサービス中の割り込み要求を追跡する。これは２つのビットベクトルを用いてしばしば行われ、それぞれのビットは個々の割り込み源を表す。第１のビットベクトルは割り込み要求レジスタ５１８と呼ばれ、第２のビットベクトルは割り込みサービスレジスタ５２０と呼ばれる。割り込みコントローラー５１６が割り込みの要求を受信すると、割り込み要求レジスタ５１８内の対応ビットをセットする。割り込みコントローラー５１６が割り込みをプロセッサー５１０、５１２、または５１４に送信すると、割り込みコントローラー５１６は割り込み要求レジスタ５１８内の対応ビットをクリアして、割り込みサービスレジスタ５２０内の対応ビットをセットする。割り込みコントローラー５１６がＥＯＩを受信すると、割り込みコントローラー５１６は、対応する割り込みがもはや行われていないことを知り、従って割り込みサービスレジスタ５２０内の対応ビットをクリアする。この時点で、割り込みコントローラー５１６は割り込み要求レジスタ５１８をスキャンして、まだサービスされていない、要求された最高優先度の割り込みを決定する。上記割り込みの優先度がサービス中の最高優先度の割り込みより高い場合、割り込みコントローラーはサービス中の低優先度の割り込みの割り込みサービスルーチンに割り込む。

【００３９】

従来の割り込み要求のライフサイクルを図６に示す。装置またはソフトウェアは割り込み要求（「ＩＲＱ」）をアサートすることで処理を開始する（６０２）。割り込み要求を多様な情報源により生成することができる。限定するためではなく例として、情報源にはキーボード、マウス、音源カード、モデム、通信ポート、時間測定装置、およびソフトウェア命令を含めることができる。「レベルトリガ」割り込みでは、割り込みを知らせたい装置は、割り込み要求の電圧を「アクティブ」と定義される規定のレベルまで上げ、割り込みが処理されるまでそのレベルに保つ。「エッジトリガ」割り込みでは、割り込み要求は割り込み要求ラインのレベル遷移により送信され、割り込みを送信したい装置は割り込み要求ラインにパルスを駆動して、その後ラインを休止状態に戻す。

【００４０】

割り込みコントローラーがＩＲＱを検出すると（６０４）、割り込みコントローラーは、ＩＲＱが現在サービス中のどの割り込みよりも高い優先度を持つかどうかを判定する（

６０６）。この判定は、割り込みサービスレジスタ５２０（図５）を検証することで可能である。ＩＲＱの検出時により優先度が高い割り込みがサービス中である場合、割り込みコントローラーは割り込み要求レジスタ内の対応ビットのフラグを立て、保留されたＩＲＱを記録する（６０８）。要求された割り込みがサービス中のどの割り込みよりも高い優先度を有する場合、割り込みコントローラーは割り込みサービスレジスタ内の対応ビットのフラグを立て（６１０）、対応する割り込みサービスルーチンを実行するよう適切なプロセッサーに知らせる（６１２）。割り込みサービスルーチンの実行完了後（６１４）、プロセッサーは、一般にＩ／Ｏポートまたはメモリーマップドレジスタを読み書きすることで、割り込みが処理されたことを割り込み装置に示すことができる（６１６）。プロセッサーは、割り込みコントローラーにＥＯＩ命令を送信することで、割り込みが処理されたことを知らせる（６１８）。当該ＥＯＩ命令は、一般にレジスタ読み書きなどの、Ｉ／ＯポートまたはメモリーマップドＩ／Ｏアクセスを通して送信される。割り込みコントローラーはＥＯＩ命令を処理し、割り込みサービスレジスタ内の対応するフラグをクリアする（６２０）。ＥＯＩ命令は、割り込みコントローラーに、保留された低優先度の割り込みがこの時点で送信できることを伝える。例えば、高優先度のキーボード割り込みが処理されている間にネットワークパケットが到着して割り込みを引き起こす場合に、割り込みコントローラーにより高優先度のキーボード割り込みが完全に処理されるまで、ネットワークパケットの割り込み要求を保留しておくことができる。ＥＯＩ命令の処理には、物理割り込みコントローラーに対して何十または何百ものサイクルを要し、仮想割り込みコントローラーに対しては何千ものサイクルを要する可能性がある。

【００４１】

図７は、割り込み優先度の一般的な概念を示す例であり、限定を意図するものではない。時刻t_1に優先度１０の割り込み源が、割り込みを要求すると仮定する（７０２）。割り込みコントローラーはプロセッサーに割り込み、当該プロセッサーがその割り込み源に関連する割り込みサービスルーチンを呼び出す（７０４）。ここで、優先度１０の割り込みに対するＩＳＲが処理を完了する前の時刻t_2に、優先度２００の割り込み源が、同一プロセッサー宛ての割り込みを要求すると仮定する（７０６）。割り込みコントローラーは再度プロセッサーに割り込み、優先度２００の割り込みに対するＩＳＲ７０８の実行を開始し、優先度１０の割り込みに対するＩＳＲを一時停止する（７１０）。優先度２００のＩＳＲ７０８が完了する前の時刻t_3に、第３の割り込み源が、優先度５０で割り込みを要求すると仮定する（７１２）。割り込みコントローラーは、プロセッサーが優先度２００のＩＳＲの実行を時刻t_4で完了するまで、この割り込みの送信を保留する（７１４）。優先度５０のＩＳＲは、優先度２００のＩＳＲ７０８が完了した後に呼び出される（７１６）。優先度５０のＩＳＲ７１６が完了すると、時刻t_5で優先度１０のＩＳＲを再開し（７１８）、時刻t_6で完了する。

【００４２】

仮想割り込みコントローラー

仮想環境では、プロセッサーおよび割り込みコントローラーを仮想化することができる。これは、ソフトウェア（例えば、ＶＭＭ）とハードウェアが提供する仮想化支援機構との組合せにより行われる。一般的な構成では、ＥＯＩ命令はＶＭＭによりエミュレートされる。これは、ＥＯＩポートまたはレジスタへのアクセスを遮断することで行われる。この遮断によりＶＭＭ内で、ＥＯＩに応答して物理割り込みコントローラーと同一の機能を実行するソフトウェアハンドラが呼び出される。遮断およびソフトウェアハンドラの組合せにより、何千何万ものサイクルが必要とされる可能性がある。これにより、仮想環境内での実行時にＩＳＲに対して大量のオーバヘッドが加わる。

【００４３】

一般にＶＭＭは割り込みを受け入れ、仮想割り込みとしてゲストオペレーティングシステムにリダイレクトする。割り込みを様々な情報源から生成することができ、当該情報源には限定するためではなく例として、物理ハードウェア装置、ハードウェア装置をエミュレートするパーティション、メッセージを送りたい、またはイベントを別のパーティショ

ンに送信したいパーティション、またはパーティションに信号を送りたいＶＭＭが含まれる。ＶＭＭは、一般に割り込みを受け入れた後に、ＥＯＩ命令を物理割り込みコントローラーに発行する。レベルトリガ割り込みに対して、ゲストオペレーティングシステム内のＩＳＲがＥＯＩ命令を仮想割り込みコントローラーに対して実行および発行するまで、ＥＯＩ命令を発行するのは、一般に安全ではない。ＡＰＩＣのような或る物理割り込みコントローラーは、サービス中の最高優先度の割り込みに対してのみＥＯＩを発行することを可能とする。しかしながら或る状況では、ＶＭＭは、サービス中の最高優先度の割り込みではない割り込みに、選択的にＥＯＩを発行する必要がある場合がある。例として、最初の割り込みの優先度が低い２つのレベルトリガ割り込みが次々と到着する際に生ずることを考える。ＶＭＭは第１の割り込みを受入れ、ゲストオペレーティングシステムにリダイレクトする。ゲスト内のＩＳＲがＥＯＩ命令を発行する前に、第２の割り込みが到着する。その後、ゲストオペレーティングシステムが、第１の割り込みにＥＯＩ命令を発行する。この場合、ＶＭＭは第１の割り込みにＥＯＩを発行することはできない。なぜならば、最高優先度の割り込みが既にサービス中であり、ＥＯＩ命令を物理割り込みコントローラーに発行すると、その最高優先度の割り込みにＥＯＩを発行することになるからである。

【００４４】

本明細書の開示によると、ゲストオペレーティングシステムはいくつかの割り込み源を「自動ＥＯＩ」としてプログラムすることができる。割り込み源を自動ＥＯＩとマークすると、従来の割り込み優先度の振る舞いが修正される。自動ＥＯＩ割り込みにより、他の割り込みの送信はブロックされない。従って、他の自動ＥＯＩ割り込みを含む任意の他の割り込みがその関連するＩＳＲの実行に割り込み可能という点で、実質的に自動ＥＯＩ割り込みは、最低優先度の割り込みと同様に振舞う。

【００４５】

自動ＥＯＩ割り込みが送信されると、割り込みサービスレジスタ内の自動ＥＯＩ割り込みに関連するビットは、即座にクリアされる。効果的に、自動ＥＯＩ割り込みが送信された時点で、仮想割り込みコントローラーがＥＯＩを自動生成する。自動ＥＯＩ割り込みを用いて、割り込み源が自身を管理し、前の割り込みが処理されたことを知るまで後続の割り込みを要求しないことが望ましい。そうでなければ、それぞれの後続の割り込みが前のＩＳＲに割り込み、プロセッサーのスタックをオーバーフローさせる可能性がある。

【００４６】

一実施形態では、自動ＥＯＩのプロパティを、ＳＩＮＴ（synthetic interrupt source）に関連する仮想レジスタ内で指定する。仮想レジスタのフォーマットは以下の通りである。

【００４７】

【表１】

ビット	説明	属性
６３：１８	予約済み（RsvdP）（値は変更不可）	読み取り／書き込み
１７	自動ＥＯＩ 割り込み送信時に暗黙的なＥＯＩ発行を行う場合にセット	読み取り／書き込み
１６	ＳＮＩＴをマスクする場合にセット	読み取り／書き込み
１５：８	予約済み（RsvdP）（値は変更不可）	読み取り／書き込み
７：０	ベクトル	読み取り／書き込み

【００４８】

仮想プロセッサーの生成時に、全てのＳＩＮＴレジスタのデフォルト値は、０ｘ０００００００００１００００である。このように、全てのＳＩＮＴはデフォルトでマスクされる。ゲストは、適切なベクトルでプログラミングし第１６ビットをクリアすることでそれらのマスクを取らなければならない。

【００４９】
自動ＥＯＩフラグは、割り込みを仮想プロセッサーに送信するとき暗黙的なＥＯＩ発行をＶＭＭが行わなければならないことを示す。さらに、ＶＭＭは、仮想割り込みコントローラー内のサービス中のレジスタにある対応フラグを自動的にクリアする。ゲストがこの振る舞いを可能とする場合、ゲストはその割り込みサービスルーチン内で明示的なＥＯＩ発行を行ってはならない。
【００５０】
図８は、本明細書の開示に従う自動ＥＯＩ割り込み要求のライフサイクルを示す。装置またはソフトウェアは、ＩＲＱをアサートすることで処理を開始する（８０２）。割り込みコントローラーは、ＩＲＱを検出すると（８０４）、ＩＲＱが、現在サービス中とマークされているどの割り込みよりも高い優先度を持つかどうかを判定する（８０６）。この判定は、割り込みサービスレジスタ５２０（図５）を検証することで可能である。ＩＲＱの検出時に高優先度の割り込みがサービス中とマークされている場合、割り込みコントローラーは割り込み要求レジスタ内の対応ビットのフラグを立て、その保留中の要求を記録する（８０８）。要求された割り込みがサービス中とマークされたどの割り込みよりも高い優先度を有する場合、割り込みコントローラーは適切なプロセッサーに、対応する割り込みサービスルーチンを実行するよう伝え（８１０）、割り込みサービスレジスタ内の対応ビットをクリアする（８１２）。プロセッサーが割り込みサービスルーチンの実行を完了した後（８１４）、一般にＩ／Ｏポートまたはメモリーマップドレジスタを読み書きすることで、プロセッサーは割り込み装置に割り込みが処理されたことを伝えることができる（８１６）。
【００５１】
従って、自動ＥＯＩ割り込み要求により、プロセッサーが明示的なＥＯＩ命令を発行する必要はない。割り込みをプロセッサーに送信したとき割り込みサービスレジスタ内の対応ビットはクリアされているので（８１２）、自動ＥＯＩ割り込みにより他の割り込みの送信がブロックされることはない。割り込みの送信時にその割り込みに対して効率的にＥＯＩが発行されるので、ＥＯＩ命令の処理に通常必要な計算サイクルは使用されない。
【００５２】
仮想割り込みコントローラーを使用することで、物理割り込みコントローラーがそのような機能をサポートしないときに、ＶＭＭは物理割り込みに対して選択的にＥＯＩを発行することができる。これは、保留ＥＯＩのリスト、即ち、後にＥＯＩを発行する必要がある割り込みを保持することで実現することができる。例えば、ゲストオペレーティングシステムがＥＯＩ命令を発行すると、ＶＭＭは、ＥＯＩが発行されている割り込みが本当に物理割り込みコントローラー内でサービス中の最高優先度の割り込みであるかどうかをチェックすることができる。最高優先度の割り込みでない場合、ＶＭＭは、単に割り込みを保留ＥＯＩのリストに追加する。他方、ＥＯＩを発行されている割り込みがサービス中の最高優先度の割り込みである場合、ＶＭＭは現在の割り込みにＥＯＩを発行するだけでなく、保留ＥＯＩのリストにある他の割り込みにもＥＯＩを発行する。
【００５３】
パーティション間メッセージング
ＶＭＭの中には、割り込みをパーティション間メッセージングの基礎として用いるものもある。パーティションはＶＭＭが置いた分離境界であり、仮想マシンに対する「コンテナ」である。或るパーティション内で実行されるソフトウェアが同一マシン上の第２のパーティション内で実行されているソフトウェアと通信する必要がある場合、パーティション間メッセージを用いて通信を行うことができる。これらのメッセージは、一般に少量のペイロードを含む。例えば、或る公知のハイパーバイザの場合、メッセージは、最大２４０バイトのメッセージペイロード、および１６バイトのヘッダを含むことができる。メッセージが送信されると、宛先パーティションに関連する仮想プロセッサーが実行可能となるまで、そのメッセージはハイパーバイザにより待ち行列に入れられる。その時点で、ハイパーバイザは割り込みをその仮想プロセッサーに送信することができる。これにより、

対応するＩＳＲが呼び出される。ＩＳＲはメッセージの読み取り、およびその内容に対処する責任を有する。上述のように、ＩＳＲは、一般に割り込みの処理後にその割り込みに対して「ＥＯＩ発行」しなければならない。この場合、ＥＯＩはメッセージの読み取り後に送信される。パーティション間メッセージングは、可能な限り高速でなければならない。従来のＥＯＩ機構では、好ましくないオーバヘッドを加え、恐らく何万ものサイクルを使用しないと、メッセージが読み出されたこと、および後続のメッセージならびに低優先度の保留された割り込みが送信可能であることを仮想割り込みコントローラーに通知することができない。

【００５４】

本明細書の開示によると、パーティション間メッセージが処理されたことを伝えるための、従来のＥＯＩ機構のオーバヘッドを回避することができる。一実施形態では、メッセージスロットをそれぞれのＳＩＮＴに与え、メッセージのレイアウトを以下の表１で記述するデータ構造により定義する。

表１

```
typedef struct
 {
      UINT8 MessagePending:l;
      UINT8 Reserved:7;
  } HV_MESSAGE_FLAGS;

typedef struct
{
      HV_MESSAGE_TYPE MessageType;
      UINT8PayloadSize;
      HV_MESSAGE_FLAGS MessageFlags
      UINT8Reserved[2];
      union
      {
          HV_PARTITION_ID Sender;
          HV_PORT_ID Port;
       };
} HV_MESSAGE_HEADER;

#defme HV_MESSAGE_MAX_PAYLOAD_BYTE_COUNT 240
#defme HV_MESSAGE_MAX_PAYLOAD_QWORD_COUNT 30

typedef struct
{
      HV_MESSAGE_HEADER Header;
      UINT64
Payload[HV_MESSAGE_MAX_PAYLOAD_QWORD_COUNT];
 } HV_MESSAGE;
```

【００５５】

図９は、本明細書の開示に従ってプロセッサー間メッセージを処理する実施形態を示すフローチャートである。送信プロセッサーは、自動ＥＯＩと指定した特定のＳＩＮＴに対応するプロセッサー間メッセージを送る（９０２）。ＶＭＭはメッセージをメッセージ待ち行列に追加し（９０４）、特定のＳＩＮＴに対応するメッセージスロットが空かどうかを判定する（９０６）。前のメッセージがメッセージスロット内にまだ存在する場合、ＶＭＭはスロットにあるメッセージのヘッダ内のメッセージ保留ビットをセットする（９０８）。メッセージスロットが空である場合、ＶＭＭはメッセージをメッセージスロットに

コピーし（９１０）、特定のＳＩＮＴに関連する割り込みを受信プロセッサーに送信する（９１２）。受信プロセッサー上で実行されているゲストＯＳがＳＩＮＴに関連する割り込みを受信すると、そのＩＳＲはメッセージを対応するメッセージスロットから読み取り、メッセージタイプおよびペイロードに基づいて動作を行う（９１４）。メッセージの処理が完了すると、ＩＳＲはメッセージタイプをクリアする（９１６）。例えば、表１に定義したデータ構造によると、ＩＳＲは特定の値をＨＶ＿ＭＥＳＳＡＧＥ＿ＴＹＰＥに書き込むことで、メッセージタイプをクリアすることができる。ＩＳＲは次いで、丁度処理したメッセージのメッセージ保留ビットを調べる（９１８）。メッセージ保留ビットがセットされていない場合、これはメッセージスロットに対する待ち行列にメッセージがないことを意味し、ＩＳＲが必要とする動作はない（９２０）。これが最もよくある場合であろう。特に、ＩＳＲがＥＯＩ命令を送信する必要はなく、従って大量の計算オーバヘッドが回避される。

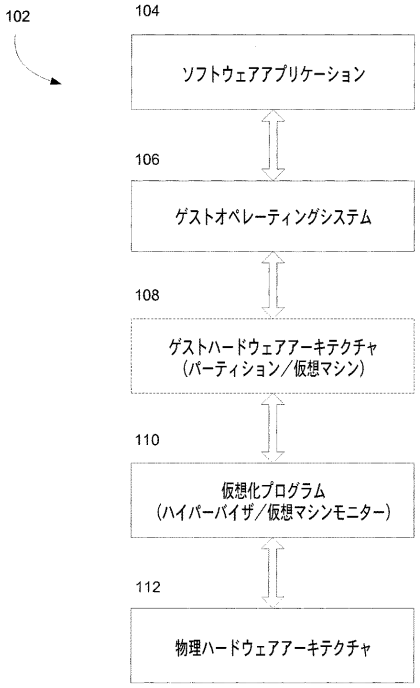
【００５６】

メッセージ保留ビットがセットされている場合、ＩＳＲは仮想割り込みコントローラーにメッセージ終了（「ＥＯＭ」）命令を送信し（９２２）、待ち行列に入れたメッセージの送信を再試行するようＶＭＭに伝える。ＥＯＭ命令の計算コストは大まかにＥＯＩ命令と同じであるが、ＥＯＭは、追加のメッセージがメッセージスロットに対する待ち行列に入れられる稀な場合にのみ送信される。従って、プロセッサー間メッセージを処理する平均的なコストは大幅に削減される。

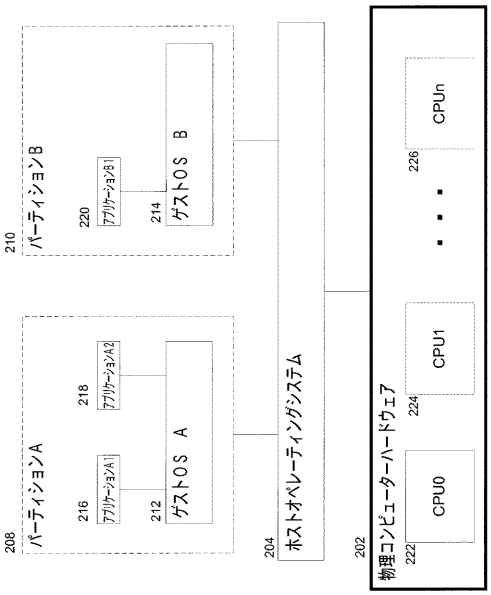
【００５７】

本開示を様々な実施形態に関連して説明し、様々な図面で示したが、本開示から逸脱することなく、同様な態様を使用し、または開示した実施形態の記載された態様に対して修正および追加を行って本開示の同一機能を実行することができることは理解される。例えば、本開示の様々な態様において、仮想環境における割り込み処理の動作効率を向上させる機構を開示した。しかしながら、これら説明した態様と同等な他の機構も、本明細書の教示により考慮されている。従って、本開示はどの態様にも限定されず、添付の特許請求の範囲に従って広範囲に解釈されるべきである。

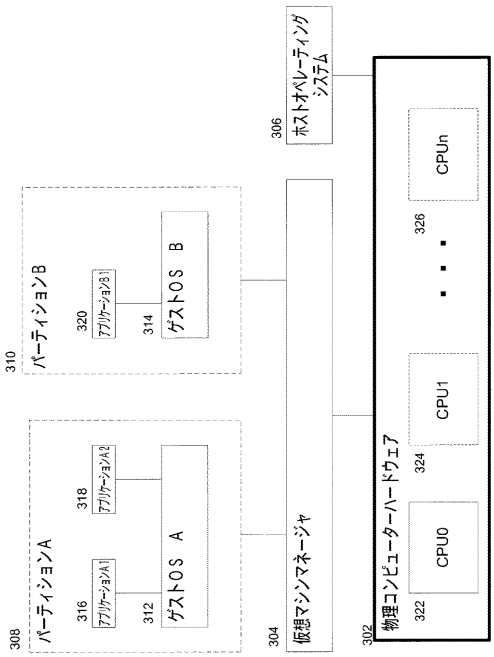
【図１】


102
104
ソフトウェアアプリケーション
106
ゲストオペレーティングシステム
108
ゲストハードウェアアーキテクチャ
（パーティション／仮想マシン）
110
仮想化プログラム
（ハイパーバイザ／仮想マシンモニター）
112
物理ハードウェアアーキテクチャ

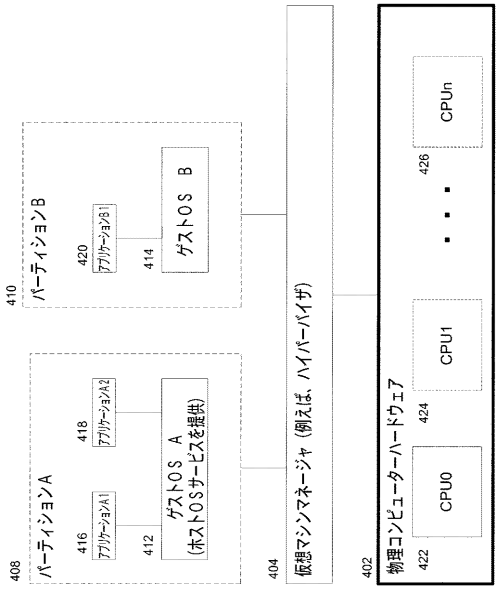
【図２】


パーティションB
210
220
アプリケーションB1
214
ゲストOS B
パーティションA
208
216
アプリケーションA1
218
アプリケーションA2
212
ゲストOS A
ホストオペレーティングシステム
204
物理コンピューターハードウェア
202
222
CPU0
224
CPU1
226
CPUn

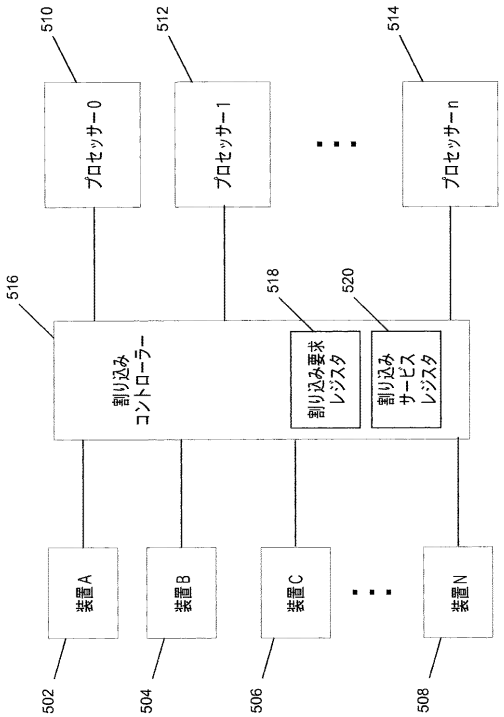
【図３】


パーティションB
310
320
アプリケーションB1
314
ゲストOS B
パーティションA
308
316
アプリケーションA1
318
アプリケーションA2
312
ゲストOS A
仮想マシンマネージャ
304
ホストオペレーティングシステム
306
物理コンピューターハードウェア
302
322
CPU0
324
CPU1
326
CPUn

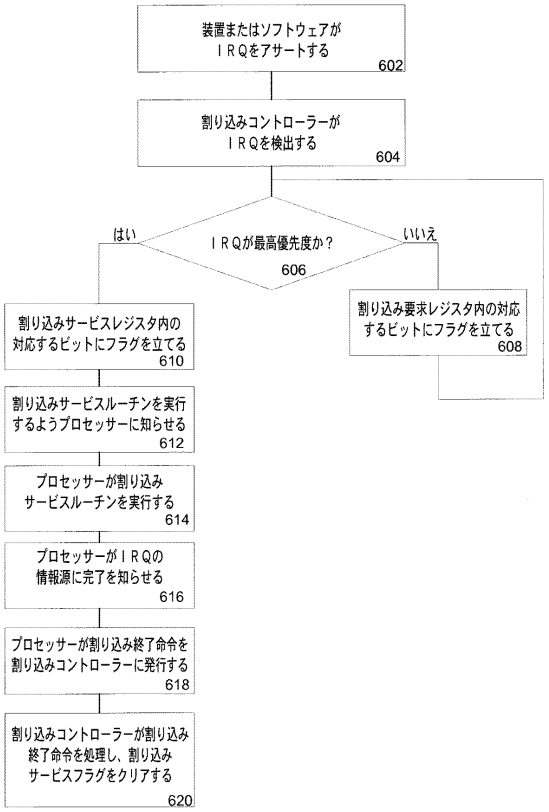
【図４】


パーティションB
410
420
アプリケーションB1
414
ゲストOS B
パーティションA
408
416
アプリケーションA1
418
アプリケーションA2
412
ゲストOS A
（ホストOSサービスを提供）
仮想マシンマネージャ（例えば、ハイパーバイザ）
404
物理コンピューターハードウェア
402
422
CPU0
424
CPU1
426
CPUn

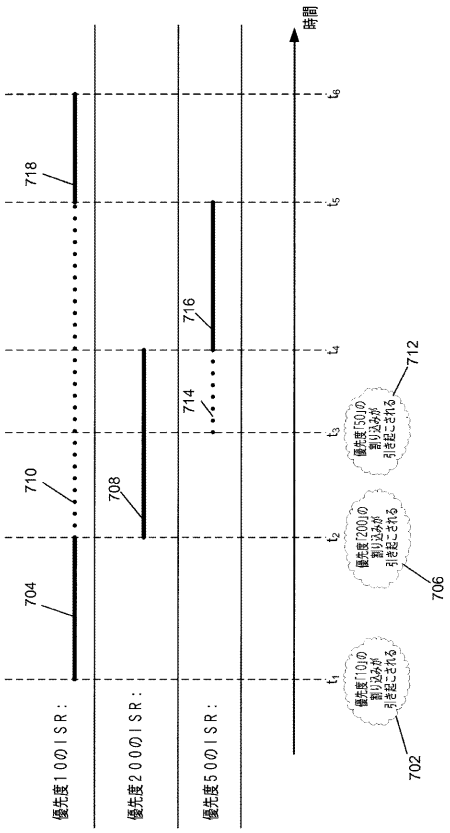
【図５】


510
512
514
プロセッサー0
プロセッサー1
プロセッサーn
516
518
520
割り込み
コントローラー
割り込み要求
レジスタ
割り込み
サービス
レジスタ
装置A
装置B
装置C
装置N
502
504
506
508

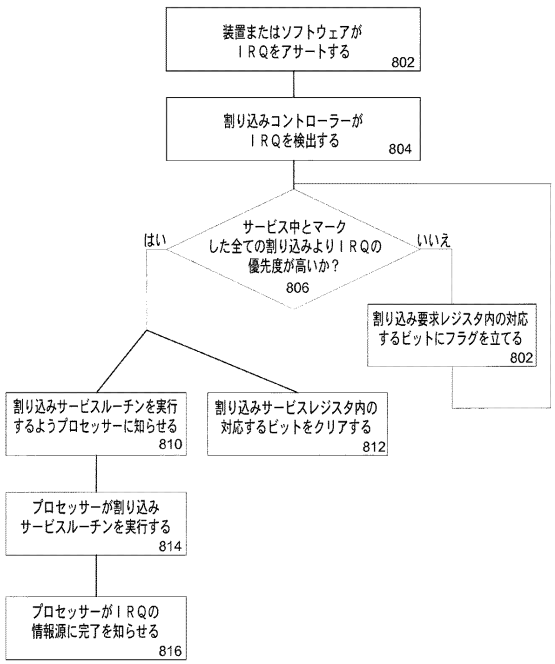
【図６】


装置またはソフトウェアが
ＩＲＱをアサートする
602
割り込みコントローラーが
ＩＲＱを検出する
604
はい
ＩＲＱが最高優先度か？
606
いいえ
割り込み要求レジスタ内の対応
するビットにフラグを立てる
608
割り込みサービスレジスタ内の
対応するビットにフラグを立てる
610
割り込みサービスルーチンを実行
するようプロセッサーに知らせる
612
プロセッサーが割り込み
サービスルーチンを実行する
614
プロセッサーがＩＲＱの
情報源に完了を知らせる
616
プロセッサーが割り込み終了命令を
割り込みコントローラーに発行する
618
割り込みコントローラーが割り込み
終了命令を処理し、割り込み
サービスフラグをクリアする
620

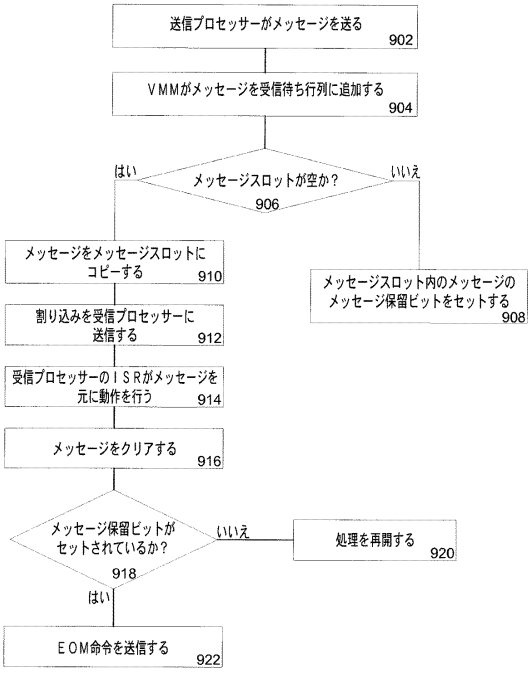
【図７】


時間
t6
t5
t4
t3
t2
t1
718
716
714
712
710
708
706
704
702
優先度100のISR：
優先度200のISR：
優先度500のISR：
優先度500の割り込みが引き起こされる
優先度200の割り込みが引き起こされる
優先度100の割り込みが引き起こされる

【図８】


装置またはソフトウェアが
ＩＲＱをアサートする
802
割り込みコントローラーが
ＩＲＱを検出する
804
はい
サービス中とマーク
した全ての割り込みよりＩＲＱの
優先度が高いか？
806
いいえ
割り込み要求レジスタ内の対応
するビットにフラグを立てる
802
割り込みサービスルーチンを実行
するようプロセッサーに知らせる
810
割り込みサービスレジスタ内の
対応するビットをクリアする
812
プロセッサーが割り込み
サービスルーチンを実行する
814
プロセッサーがＩＲＱの
情報源に完了を知らせる
816

【図９】


送信プロセッサーがメッセージを送る
902
VMMがメッセージを受信待ち行列に追加する
904
はい
メッセージスロットが空か？
906
いいえ
メッセージをメッセージスロットにコピーする
910
メッセージスロット内のメッセージのメッセージ保留ビットをセットする
908
割り込みを受信プロセッサーに送信する
912
受信プロセッサーのＩＳＲがメッセージを元に動作を行う
914
メッセージをクリアする
916
メッセージ保留ビットがセットされているか？
918
いいえ
処理を再開する
920
はい
ＥＯＭ命令を送信する
922

フロントページの続き

(72)発明者　エリック　ピー．トラウト
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　インターナショナル　パテンツ内
(72)発明者　シューヴァブラタ　ガングリー
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　インターナショナル　パテンツ内
(72)発明者　ルネ　アントニオ　ベガ
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　インターナショナル　パテンツ内

審査官　鈴木　修治

(56)参考文献　特開平０５－０４０６４３（ＪＰ，Ａ）
特開平０１－０９３８３０（ＪＰ，Ａ）
特開平０６－０８３６４２（ＪＰ，Ａ）
特開平０９－０９７１８４（ＪＰ，Ａ）
特開平０１－１９１２３４（ＪＰ，Ａ）
特開平０３－０６５７３４（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F　9/46
G06F　9/48