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

特許第5532793号
(P5532793)

(45)発行日　平成26年6月25日(2014.6.25)　(24)登録日　平成26年5月9日(2014.5.9)

(51)Int.Cl.　　F I
G06F 9/46 (2006.01)　　G06F 9/46 350
G06F 9/54 (2006.01)　　G06F 9/46 480A

請求項の数 2　(全 16 頁)

(21)出願番号　特願2009-222527(P2009-222527)
(22)出願日　平成21年9月28日(2009.9.28)
(65)公開番号　特開2011-70526(P2011-70526A)
(43)公開日　平成23年4月7日(2011.4.7)
審査請求日　平成24年6月5日(2012.6.5)

(73)特許権者　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　100070150
弁理士　伊東　忠彦
(74)代理人　100146776
弁理士　山口　昭則
(72)発明者　兒島　尚
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　中田　正弘
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

最終頁に続く

(54)【発明の名称】プログラム及び通信制御方法

(57)【特許請求の範囲】
【請求項１】
コンピュータに、
ハイパーバイザ上で動作する制御仮想計算機を前記ハイパーバイザ上に起動する制御仮想計算機起動ステップと、
前記ハイパーバイザ上で動作する仮想計算機を前記ハイパーバイザ上に起動する仮想計算機起動ステップと、
前記ハイパーバイザが、前記仮想計算機又は前記制御仮想計算機からの通信要求を監視して、単純アクセス設定に基づき、前記仮想計算機と前記制御仮想計算機との間の通信のみを許可するようにアクセス制御する単純アクセス制御ステップと、
前記制御仮想計算機が、前記ハイパーバイザから、転送先である仮想計算機を指定した前記仮想計算機からの通信要求を受信する受信ステップと、
前記制御仮想計算機が、前記受信した通信要求に含まれるメッセージの、メッセージ内容とアクセスの可否を対応付けた高度アクセス制御設定に基づいてアクセスの可否を判定する判定ステップと、
前記制御仮想計算機が、前記判定ステップにより許可された通信要求を、前記転送先である仮想計算機に対して送信するようにアクセス制御する高度アクセス制御ステップと、を実行させるためのプログラム。
【請求項２】
コンピュータによって実行される通信制御方法であって、

前記コンピュータが、
ハイパーバイザ上で動作する制御仮想計算機を前記ハイパーバイザ上に起動する制御仮想計算機起動ステップと、
前記ハイパーバイザ上で動作する仮想計算機を前記ハイパーバイザ上に起動する仮想計算機起動ステップと、
前記ハイパーバイザが、前記仮想計算機又は前記制御仮想計算機からの通信要求を監視して、単純アクセス設定に基づき、前記仮想計算機と前記制御仮想計算機との間の通信のみを許可するようにアクセス制御する単純アクセス制御ステップと、
前記制御仮想計算機が、前記ハイパーバイザから、転送先である仮想計算機を指定した前記仮想計算機からの通信要求を受信する受信ステップと、
前記制御仮想計算機が、前記受信した通信要求に含まれるメッセージの、メッセージ内容とアクセスの可否を対応付けた高度アクセス制御設定に基づいてアクセスの可否を判定する判定ステップと、
前記制御仮想計算機が、前記判定ステップにより許可された通信要求を、前記転送先である仮想計算機に対して送信するようにアクセス制御する高度アクセス制御ステップと、
を実行する通信制御方法。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、プログラム及び通信制御方法に係り、特に仮想化環境において実行された複数の仮想計算機を制御するプログラム及び通信制御方法に関する。

【背景技術】

【０００２】

近年、ＣＰＵの高速化や記憶媒体の高容量化，ネットワークの高速化等のコンピュータ技術の進化により、ハードウェアを抽象化する仮想化技術が、ハードウェア利用率の向上やＯＳ間のセキュリティの向上、複製やバックアップの容易性による効率的な管理などを目的として広く利用されている。

【０００３】

仮想化技術はVMware ESXi（登録商標）、Xen（登録商標）、Hyper-V（登録商標）などのハイパーバイザ（Hypervisor）と呼ばれる小さなプログラムにより実現される（例えば特許文献１参照）。

【０００４】

ハイパーバイザは一般的なＯＳとハードウェアとの間に入り、ハードウェアの機能を仮想化することにより複数のＯＳを仮想的に一つのハードウェア上で動作させることを特徴とする。以下の説明では、ハイパーバイザ上で動作するＯＳ等のプログラムをＶＭ（仮想機械：Virtual Machine）と呼ぶ。

【０００５】

ハイパーバイザ上で動作するＶＭが増加すると、ハイパーバイザ上には互いに信頼関係のないＶＭ群が増加していく。したがって、ハイパーバイザ上のＶＭ間の通信にはアクセス制御が必要となる。ＶＭ間の通信には、一般的に、ＶＭ間割り込みと共有メモリとの２種類がある。

【０００６】

ＶＭ間割り込みは、一方のＶＭから他方のＶＭにイベントの発生を通知する為のものである。ＶＭ間割り込みは、宛先とメッセージとをハイパーコール（Hypercall）によりハイパーバイザに通知する。ハイパーコールはハイパーバイザの機能を利用するためのＶＭ向けのインターフェースの総称である。ハイパーバイザは宛先のＶＭにメッセージを送信する。

【０００７】

共有メモリは、ＶＭ同士で特定のメモリ領域を共有することでデータの共有を実現するものである。共有メモリを利用する場合、一般的な方法では一方のＶＭが特定のメモリ領

域をハイパーコールにより共有可能に設定し、他方のＶＭが同じくハイパーコールにより共有可能なメモリ領域を共有状態に設定する。

【０００８】

通常、ＶＭ間の通信はＶＭ間割り込みによってイベントを発生させ、そのイベントが正しく処理された後に、共有メモリによりデータをやり取りする方法が一般的である。このため、ＶＭ間の通信のアクセス制御はＶＭ間割り込みに対して行われることが多い。例えばＶＭ間の通信のアクセス制御は、特定のＶＭからのＶＭ間割り込みを拒絶したり、特定の種類のメッセージのみを許可したりといった制御が考えられる。

【０００９】

一般的なハイパーバイザではＶＭ自身が自らに割り込みを送信可能な他のＶＭのＩＤを設定できるようになっている。しかし、この設定はＶＭのＩＤに基づき、送信可能・不可能を制御するだけであり、特定の種類のメッセージのみを許可するといった制御を行うことができない。特定の種類のメッセージのみを許可するといった制御を行う場合は別の仕組みが必要であった。

【００１０】

以下では、ＶＭ自身が自らに割り込みを送信可能な他のＶＭのＩＤを設定することによるＶＭ間の通信のアクセス制御を単純アクセス制御と呼び、特定の種類のメッセージのみを許可するといった制御まで考慮するアクセス制御を高度アクセス制御と呼ぶ。例えば高度アクセス制御を行う場所としてはハイパーバイザが知られている。

【００１１】

ハイパーバイザには高度アクセス制御を支援する機能を備えているものがあった。例えばハイパーバイザの一例のXen（登録商標）には、ＸＳＭ（Xen Security Modules）と呼ばれる仕組みが搭載されており、予め定義されたフック関数に独自の高度アクセス制御のモジュールを追加できるようになっている。以下では、ハイパーバイザで行う高度アクセス制御をハイパーバイザ高度アクセス制御と呼ぶ。

【先行技術文献】

【特許文献】

【００１２】

【特許文献１】特開２００９－２６１１７号公報

【発明の概要】

【発明が解決しようとする課題】

【００１３】

しかしながら、高度アクセス制御を支援する機能は、全てのハイパーバイザに用意されている訳ではない。したがって、高度アクセス制御を支援する機能が用意されていないハイパーバイザはハイパーバイザ高度アクセス制御を行うことができなかった。

【００１４】

特に一般的なＰＣ向けのハイパーバイザでなく、組み込み向けのハイパーバイザの場合は計算能力やメモリ容量の制限から必要最低限の機能で動作することが求められる。このため、ハイパーバイザ高度アクセス制御のような機能は、省略されることが多いと考えられる。

【００１５】

本発明の一実施形態は、上記の点に鑑みなされたもので、仮想化環境において実行された複数の仮想計算機間の通信を容易且つ柔軟に制御できるプログラム及び通信制御方法を提供することを目的とする。

【課題を解決するための手段】

【００１６】

上記課題を解決する為、本発明の一実施形態は、コンピュータに、ハイパーバイザ上で動作する制御仮想計算機を前記ハイパーバイザ上に起動する制御仮想計算機起動ステップと、前記ハイパーバイザ上で動作する仮想計算機を前記ハイパーバイザ上に起動する仮想計算機起動ステップと、前記ハイパーバイザが、前記仮想計算機又は前記制御仮想計算機

からの通信要求を監視して、単純アクセス設定に基づき、前記仮想計算機と前記制御仮想計算機との間の通信のみを許可するようにアクセス制御する単純アクセス制御ステップと、前記制御仮想計算機が、前記ハイパーバイザから、転送先である仮想計算機を指定した前記仮想計算機からの通信要求を受信する受信ステップと、前記制御仮想計算機が、前記受信した通信要求に含まれるメッセージの、メッセージ内容とアクセスの可否を対応付けた高度アクセス制御設定に基づいてアクセスの可否を判定する判定ステップと、前記制御仮想計算機が、前記判定ステップにより許可された通信要求を、前記転送先である仮想計算機に対して送信するようにアクセス制御する高度アクセス制御ステップと、を実行させるためのプログラムである。

【００１７】

なお、本発明の一実施形態の構成要素、表現または構成要素の任意の組合せを、方法、装置、システム、コンピュータプログラム、記録媒体、データ構造などに適用したものも本発明の態様として有効である。

【発明の効果】

【００１８】

上述の如く、本発明の一実施形態によれば、仮想化環境において実行された複数の仮想計算機間の通信を容易且つ柔軟に制御できる。

【図面の簡単な説明】

【００１９】

【図１】本実施例におけるソフトウェアとハードウェアとの関係を表した一例のブロック図である。
【図２】本実施例におけるハードウェアの一例の構成図である。
【図３】個別高度アクセス制御を説明する為の説明図である。
【図４】ハイパーバイザ高度アクセス制御を説明する為の説明図である。
【図５】制御ＶＭで行う高度アクセス制御を説明する為の説明図である。
【図６】準備処理の手順を表した一例のフローチャートである。
【図７】単純アクセス制御設定のハイパーコールの例を表した説明図である。
【図８】単純アクセス制御設定テーブルの一例の構成図である。
【図９】ハイパーバイザへの共有メモリ設定依頼の例を表した説明図である。
【図１０】共有メモリ設定テーブルの一例の構成図である。
【図１１】ハイパーバイザへの共有メモリ使用依頼の例を表した説明図である。
【図１２】アクセス処理の手順を表した一例のフローチャートである。
【図１３】ハイパーバイザへの割り込み送信依頼の例を表した説明図である。
【図１４】割り込みに含めるメッセージの例を表した説明図である。
【図１５】メッセージ本体に含まれる情報の例を表した説明図である。
【図１６】高度アクセス制御設定の例を表した説明図である。

【発明を実施するための形態】

【００２０】

次に、本発明を実施するための形態を、以下の実施例に基づき図面を参照しつつ説明していく。

【００２１】

図１は本実施例におけるソフトウェアとハードウェアとの関係を表した一例のブロック図である。図１のブロック図は、ハードウェア１０と、ハードウェア１０上で動作するハイパーバイザ２０と、ハイパーバイザ２０上で動作する制御ＶＭ３０と、ハイパーバイザ２０上で動作する複数のＶＭ（一般ＶＭ）１～ｎとを表している。

【００２２】

複数のＶＭ１～ｎはハイパーバイザ２０による単純アクセス制御及び制御ＶＭ３０による高度アクセス制御の対象となるＶＭ（制御対象ＶＭ群）である。制御ＶＭ３０は複数のＶＭ１～ｎに対して高度アクセス制御を行う。なお、制御ＶＭ３０の詳細は後述する。

【００２３】

図２は本実施例におけるハードウェアの一例の構成図である。図２はハードウェア１０が一般的なＰＣ等のコンピュータシステムである例を表している。

【００２４】

図２のコンピュータシステムは、バスＢで相互に接続されている入力装置１１，出力装置１２，ドライブ装置１３，補助記憶装置１４，主記憶装置１５，演算処理装置１６及びインターフェース装置１７を有する。

【００２５】

入力装置１１はキーボードやマウス等である。入力装置１１は、各種信号を入力するために用いられる。出力装置１２はディスプレイ装置等である。出力装置１２は各種ウインドウやデータ等を表示するために用いられる。インターフェース装置１７はモデム，ＬＡＮカード等である。インターフェース装置１７はネットワークに接続する為に用いられるものである。

【００２６】

本実施例のハイパーバイザ２０は、図２のコンピュータシステムを制御する各種プログラムの少なくとも一部である。ハイパーバイザ２０は例えば記録媒体１８の配布やネットワークからのダウンロードなどによって提供される。ハイパーバイザ２０を記録した記録媒体１８は、ＣＤ－ＲＯＭ、フレキシブルディスク、光磁気ディスク等の様に情報を光学的，電気的或いは磁気的に記録する記録媒体、ＲＯＭ、フラッシュメモリ等の様に情報を電気的に記録する半導体メモリ等、様々なタイプの記録媒体を用いることができる。

【００２７】

また、ハイパーバイザ２０を記録した記録媒体１８がドライブ装置１３にセットされると、ハイパーバイザ２０は記録媒体１８からドライブ装置１３を介して補助記憶装置１４にインストールされる。ネットワークからダウンロードされたハイパーバイザ２０は、インターフェース装置１７を介して補助記憶装置１４にインストールされる。

【００２８】

補助記憶装置１４は、インストールされたハイパーバイザ２０を格納すると共に、必要なファイル，データ等を格納する。主記憶装置１５は、ハイパーバイザ２０の起動時に補助記憶装置１４からハイパーバイザ２０を読み出して格納する。そして、演算処理装置１６は主記憶装置１５に格納されたハイパーバイザ２０に従って、後述するような各種処理を実現している。

【００２９】

高度アクセス制御を行う場所は本実施例のように制御ＶＭ３０で行う他、ハイパーバイザ２０、各ＶＭ１～ｎが考えられる。そこで、ここでは本実施例の理解を容易とする為に各ＶＭ１～ｎで行う高度アクセス制御及びハイパーバイザ２０で行うハイパーバイザ高度アクセス制御の課題について説明する。以下では、各ＶＭ１～ｎで行う高度アクセス制御を個別高度アクセス制御と呼ぶ。

【００３０】

図３は個別高度アクセス制御を説明する為の説明図である。なお、図３ではハイパーバイザ２０上で動作するＶＭが３つの例を表している。図３のＶＭ１～３は、割り込み送信部４１，本体処理部４２，高度アクセス制御部４３，割り込み受信部４４，共有メモリ設定テーブル４５，高度アクセス制御設定テーブル４６を有する。ハイパーバイザ２０は割り込み受信部２１，単純アクセス制御部２２，割り込み送信部２３，単純アクセス制御設定テーブル２４を有する。また、図３ではＶＭ１及びＶＭ２の共有メモリ４７と、ＶＭ２及びＶＭ３の共有メモリ４８とを図示している。

【００３１】

各ＶＭ１～３は、それぞれが高度アクセス制御部４３と高度アクセス制御設定テーブル４６とを有している。各ＶＭ１～３は、高度アクセス制御部４３と高度アクセス制御設定テーブル４６とを利用することで、自らに送信されたＶＭ間割り込みを検査し、予め高度アクセス制御設定テーブル４６に設定された高度アクセス制御設定に基づき、必要な個別高度アクセス制御を行う。

【００３２】
しかし、個別高度アクセス制御は、ハイパーバイザ２０上で動作するＶＭの数が増大してくると、設定すべき高度アクセス制御設定の数が増大し、管理が煩雑になるという問題があった。管理の煩雑さは、高度アクセス制御設定のミスを誘発する原因となり、安全性が損なわれる恐れがあった。
【００３３】
図４はハイパーバイザ高度アクセス制御を説明する為の説明図である。なお、図４ではハイパーバイザ２０上で動作するＶＭが３つの例を表している。図４のＶＭ１～３は割り込み送信部４１，本体処理部４２，割り込み受信部４４，共有メモリ設定テーブル４５を有する。ハイパーバイザ２０は割り込み受信部２１，単純アクセス制御部２２，割り込み送信部２３，単純アクセス制御設定テーブル２４，フック部２５を有する。高度アクセス制御モジュール５０は、高度アクセス制御部５１，高度アクセス制御設定テーブル５２を有する。また、図４ではＶＭ１及びＶＭ２の共有メモリ４７と、ＶＭ２及びＶＭ３の共有メモリ４８とを図示している。
【００３４】
ハイパーバイザ２０はフック部２５に定義されたフック関数を利用して、高度アクセス制御モジュール５０を追加できる。ハイパーバイザ高度アクセス制御は個別高度アクセス制御と異なり、ハイパーバイザ２０側の高度アクセス制御モジュール５０で高度アクセス制御を一元管理できるので、高度アクセス制御を管理しやすい。しかし、ハイパーバイザ高度アクセス制御はハイパーバイザ２０側に高度アクセス制御を実現する機能が用意されていなければならなかった。例えば組み込み向けのハイパーバイザ等は計算能力やメモリ容量の制限から、高度アクセス制御を実現する機能が用意されていない場合も多い。
【００３５】
図５は制御ＶＭで行う高度アクセス制御を説明する為の説明図である。なお、図５ではハイパーバイザ２０上で動作するＶＭが３つの例を表している。図５のＶＭ１～３は、割り込み送信部４１，本体処理部４２，割り込み受信部４４，共有メモリ設定テーブル４５を有する。ハイパーバイザ２０は、割り込み受信部２１，単純アクセス制御部２２，割り込み送信部２３，単純アクセス制御設定テーブル２４を有する。制御ＶＭ３０は、割り込み送信部３１，高度アクセス制御部３２，割り込み受信部３３，高度アクセス制御設定テーブル３４を有する。また、図５ではＶＭ１及びＶＭ２の共有メモリ４７と、ＶＭ２及びＶＭ３の共有メモリ４８とを図示している。
【００３６】
図５では、各ＶＭ１～３がアクセスできる唯一のＶＭとして、制御ＶＭ３０を用意している。制御ＶＭ３０以外の各ＶＭ１～３間の通信は全て制御ＶＭ３０に送られ、制御ＶＭ３０から目的のＶＭ１～３に受け渡される。制御ＶＭ３０は各ＶＭ１～３間の通信を一括して高度アクセス制御できるので、ハイパーバイザ高度アクセス制御が無くても、制御ＶＭ３０による一元的な高度アクセス制御を行うことができ、高度アクセス制御設定のミスを起こりにくくできる。
【００３７】
また、高度アクセス制御を行う各ＶＭ１～３間の通信はＶＭ間割り込みとする。各ＶＭ１～３間のアクセス制御はＶＭ間割り込みに対して行えば充分であり、ＶＭ１及びＶＭ２の共有メモリ４７、ＶＭ２及びＶＭ３の共有メモリ４８に対して行う必要はない。
【００３８】
一般に、ＶＭ１及びＶＭ２の共有メモリ４７、ＶＭ２及びＶＭ３の共有メモリ４８の高度アクセス制御を行う為には、比較的大きなオーバーヘッドが生じる。しかし、ＶＭ間割り込みの高度アクセス制御を行う場合、割り込み転送のためのＶＭ間割り込みの回数が２倍に増えるものの、ＶＭ間割り込み自体のオーバーヘッドは充分に小さいため、比較的大きなオーバーヘッドが生じにくい。よって、制御ＶＭ３０で行う高度アクセス制御は少ないオーバーヘッドで各ＶＭ１～３間のアクセス制御を効率的に行うことができる。
【００３９】

また、各ＶＭ１～３間のＶＭ間割り込みは、直接の送信先である制御ＶＭ３０の識別子に加えて、実際の通信先である各ＶＭ１～３の識別子を持つ。したがって、制御ＶＭ３０は実際の通信先である各ＶＭ１～３にＶＭ間割り込みを転送できる。

【００４０】

さらに、制御ＶＭ３０は予め設定されている高度アクセス制御設定に基づき、特定のＶＭ（例えばＶＭ１）から特定のＶＭ（例えばＶＭ２）への通信を禁止する為、ユーザの望む高度アクセス制御を行うことができる。

【実施例１】

【００４１】

以下、本実施例のハイパーバイザ２０による単純アクセス制御及び制御ＶＭ３０による高度アクセス制御の処理手順について説明していく。ここでは、ＶＭ間割り込み及び共有メモリによりＶＭ間の通信を行っているＶＭ１とＶＭ２とがあり、ＶＭ１からＶＭ２に割り込みを送信する場合の制御ＶＭ３０による高度アクセス制御の例について説明する。

【００４２】

図６は準備処理の手順を表した一例のフローチャートである。ハイパーバイザ２０は事前に起動しているものとする。ステップＳ１００に進み、ハイパーバイザ２０は所定の設定ファイル（図示せず）に従い、制御ＶＭ３０を起動する。制御ＶＭ３０の起動に必要なファイルはストレージなどの補助記憶装置１４に保存されている。ハイパーバイザ２０は制御ＶＭ３０の起動に必要なファイルをストレージなどの補助記憶装置１４からロードして起動する。

【００４３】

一般的なハイパーバイザ２０は起動したＶＭに一意のＶＭ＿ＩＤを付与する。特に設定しない限り、起動されたＶＭは割り込みを一切受け付けない状態であり、どのＶＭともメモリを共有していない。ここでは制御ＶＭ３０のＶＭ＿ＩＤを「０」とする。

【００４４】

ステップＳ１０１に進み、ハイパーバイザ２０は所定の設定ファイルに従い、ＶＭ１を起動する。ハイパーバイザ２０はＶＭ１の起動に必要なファイルを補助記憶装置１４からロードして起動する。起動されたＶＭ１は割り込みを一切受け付けない状態であり、どのＶＭともメモリを共有していない。ここではＶＭ１のＶＭ＿ＩＤを「１」とする。

【００４５】

ステップＳ１０２に進み、ハイパーバイザ２０は所定の設定ファイルに従い、ＶＭ２を起動する。ハイパーバイザ２０はＶＭ２の起動に必要なファイルを補助記憶装置１４からロードして起動する。起動されたＶＭ２は割り込みを一切受け付けない状態であり、どのＶＭともメモリを共有していない。ここではＶＭ２のＶＭ＿ＩＤを「２」とする。

【００４６】

ステップＳ１０３に進み、ＶＭ１は制御ＶＭ３０（ＶＭ＿ＩＤが０のＶＭ）からの割り込みを受け付けるように、単純アクセス制御設定をハイパーバイザ２０に要求する。一般的なハイパーバイザ２０では、各ＶＭ自身が、自分への割り込みを許可する他のＶＭを設定できる。

【００４７】

通常、各ＶＭは自分への割り込みを許可する他のＶＭのＶＭ＿ＩＤをハイパーコールによりハイパーバイザ２０へ通知することで、自分への割り込みを許可する他のＶＭを指定できる。図７は単純アクセス制御設定のハイパーコールの例を表した説明図である。単純アクセス制御設定のハイパーコールは、対象のＶＭのＶＭ＿ＩＤと、割り込みを受けるか否かの情報とが指定される。

【００４８】

ハイパーバイザ２０は、ハイパーコールを行った呼び出し元のＶＭ１のＶＭ＿ＩＤである「１」と、ハイパーコールで対象のＶＭのＶＭ＿ＩＤとして指定された「０」と、割り込みを受けるか否かの情報として指定された「ＴＲＵＥ（割り込みを受ける）」とに基づいて、図８のような単純アクセス制御設定テーブル２４に単純アクセス制御を設定する。

【００４９】
図８は単純アクセス制御設定テーブルの一例の構成図である。図８の単純アクセス制御設定テーブル２４は、送信側ＶＭ＿ＩＤと受信側ＶＭ＿ＩＤとの組み合わせごとに、割り込みが可能か不可かを設定するものである。ハイパーバイザ２０はハイパーコールで指定された対象のＶＭのＶＭ＿ＩＤを送信側ＶＭ＿ＩＤ、ハイパーコールを行った呼び出し元のＶＭのＶＭ＿ＩＤを受信側ＶＭ＿ＩＤ、ハイパーコールで指定された割り込みを受けるか否かの情報を割り込みが可能か不可かの情報として、単純アクセス制御を設定する。
【００５０】
本実施例において、制御ＶＭ３０以外の全てのＶＭの割り込み設定は、制御ＶＭ３０からの割り込みを許可し、制御ＶＭ３０以外からの割り込みを許可しないように設定することを前提とする。これにより、各ＶＭ間の割り込みは必ず制御ＶＭ３０を経由する。
【００５１】
ステップＳ１０４に進み、ＶＭ２はＶＭ１と同様、制御ＶＭ３０（ＶＭ＿ＩＤが０のＶＭ）からの割り込みを受け付けるように、単純アクセス制御設定をハイパーバイザ２０に要求する。ハイパーバイザ２０は、ＶＭ２が制御ＶＭ３０からの割り込みを許可し、制御ＶＭ３０以外からの割り込みを許可しないように、単純アクセス制御を単純アクセス制御設定テーブル２４に設定する。
【００５２】
ステップＳ１０５に進み、制御ＶＭ３０はＶＭ１及びＶＭ２からの割り込みを受け付けるように、単純アクセス制御設定をハイパーバイザ２０に要求する。ハイパーバイザ２０は制御ＶＭ３０がＶＭ１及びＶＭ２からの割り込みを許可し、ＶＭ１及びＶＭ２以外からの割り込みを許可しないように、単純アクセス制御を単純アクセス制御設定テーブル２４に設定する。
【００５３】
ステップＳ１０６に進み、ＶＭ１はハイパーバイザ２０に、ＶＭ１の特定のメモリ領域をＶＭ２と共有できるように、特定のメモリ領域を共有メモリとして設定するように要求する。
【００５４】
一般的なハイパーバイザ２０では、ＶＭのメモリ領域に、ＶＭ内で一意な番号（メモリ領域の番号）を付与することになっており、共有メモリを指定する際、メモリ領域の番号を指定する。メモリ領域の番号は他のＶＭが共有メモリを実際に使用する際、ハイパーバイザ２０にメモリ領域を指定する番号として利用される。
【００５５】
ここでは、共有メモリとして設定するメモリ領域の番号を予めＶＭ１とＶＭ２とで定義しておくものとし、互いの共有メモリ設定情報として共有メモリ設定テーブル４５に保持しているものとする。
【００５６】
図９はハイパーバイザへの共有メモリ設定依頼の例を表した説明図である。共有メモリ設定依頼は、対象のメモリ領域の番号と、共有するかどうかの情報とが指定される。図１０は共有メモリ設定テーブルの一例の構成図である。共有メモリ設定テーブル４５は各ＶＭが保持する共有メモリ設定情報を表す。図１０の共有メモリ設定情報は、ＶＭ１の共有メモリとして設定されたメモリ領域の番号のリストを表している。
【００５７】
ステップＳ１０７に進み、ＶＭ２は共有メモリ設定テーブル４５に予め設定しておいたＶＭ１の共有メモリとして設定されたメモリ領域の番号に基づき、ハイパーバイザ２０にＶＭ１の共有メモリとして設定されたメモリ領域を使用できるように要求する。図１１はハイパーバイザへの共有メモリ使用依頼の例を表した説明図である。共有メモリ使用依頼は対象のＶＭのＶＭ＿ＩＤと、対象のメモリ領域の番号と、使用するかどうかの情報とが指定されている。
【００５８】

図６に示した準備処理が終了した後、ＶＭ１とＶＭ２とは共有メモリを経由してデータをやり取りすることができるようになる。共有メモリへのアクセスはＶＭ１，２の本体処理部４２が行うものとする。

【００５９】

図１２はアクセス処理の手順を表した一例のフローチャートである。ステップＳ１０８に進み、ＶＭ１はＶＭ２に割り込みを送信するため、ハイパーバイザ２０に制御ＶＭ３０への割り込みを依頼する。

【００６０】

ＶＭ１の実際の割り込みの宛先はＶＭ２であるが、制御ＶＭ３０による高度アクセス制御を実現するため、ＶＭ１は割り込みの宛先を全て制御ＶＭ３０としている。ＶＭ１はハイパーバイザ２０に通知する通常の割り込みの宛先（制御ＶＭ３０）の他、制御ＶＭ３０に通知する実際の割り込みの宛先を指定する必要がある。

【００６１】

一般的なハイパーバイザ２０では、割り込みの際に簡単なメッセージを追加することができる。そこで、ＶＭ１は割り込みの際に追加できる簡単なメッセージを利用して、実際の割り込みの宛先であるＶＭ２のＶＭ＿ＩＤを指定する。図１３はハイパーバイザへの割り込み送信依頼の例を表した説明図である。図１３の割り込み送信依頼は、制御ＶＭ３０のＶＭ＿ＩＤと、割り込みに含めるメッセージとが指定されている。

【００６２】

図１４は割り込みに含めるメッセージの例を表した説明図である。図１４の割り込みに含めるメッセージは、実際の割り込みの宛先であるＶＭ２のＶＭ＿ＩＤと、メッセージ本体とが指定されている。図１４のメッセージ本体には、様々な情報が含まれる可能性があるが、図１５のような情報が含まれるものとする。図１５はメッセージ本体に含まれる情報の例を表した説明図である。図１５のメッセージ本体に含まれる情報は、コマンド，引数１，引数２を指定するものである。

【００６３】

具体的に、図１５のメッセージ本体に含まれる情報は、ファイル削除を依頼するコマンドと、対象ファイルのディレクトリ名と、対象ファイル名とを指定することで、特定のパスのファイルの削除を依頼するものである。なお、図１４及び図１５では、割り込みに含めるメッセージ及びメッセージ本体に含まれる情報を表形式で表しているが、視覚的に理解し易く表したものである。実際の割り込みに含めるメッセージ及びメッセージ本体に含まれる情報は、例えばスペースで連結した文字列で表される。

【００６４】

ステップＳ１０９に進み、ハイパーバイザ２０はＶＭ１から制御ＶＭ３０への割り込み送信依頼を受信し、単純アクセス制御設定テーブル２４に保持されている単純アクセス制御設定に基づいて、ＶＭ１から制御ＶＭ３０への割り込みが許可されているかをチェックする。ＶＭ１から制御ＶＭ３０への割り込みが許可されていれば、ハイパーバイザ２０はステップＳ１１０に進み、ＶＭ１からの割り込みを制御ＶＭ３０に送信する。

【００６５】

一方、ＶＭ１から制御ＶＭ３０への割り込みが許可されていなければ、不正な割り込み送信依頼であるため、ハイパーバイザ２０はステップＳ１１１に進み、ＶＭ１にエラーを返して異常終了する。

【００６６】

ステップＳ１１０に続いてステップＳ１１２に進み、制御ＶＭ３０はＶＭ１からの割り込みをハイパーバイザ２０から受信する。ステップＳ１１３に進み、制御ＶＭ３０は高度アクセス制御設定テーブル３４に保持されている高度アクセス制御設定に基づいて、ＶＭ１からＶＭ２への割り込みが許可されているかをチェックする。

【００６７】

制御ＶＭ３０は高度アクセス制御設定を予め高度アクセス制御設定テーブル３４に保持しているものとする。図１６は高度アクセス制御設定の例を表した説明図である。図１６

の高度アクセス制御設定は、宛先ＶＭ，コマンド，引数１，引数２を指定する。制御ＶＭ３０は、図１５に示したメッセージ本体に含まれる情報が、図１６の高度アクセス制御設定に一致するとき、ＶＭ１からＶＭ２への割り込みが許可されていると判断する。図１６の高度アクセス制御設定は、ＶＭ２への割り込みが、引数１で指定された特定のディレクトリの配下のファイルについてファイルの削除の依頼であるときに、ＶＭ１からＶＭ２への割り込みを許可するものである。つまり、制御ＶＭ３０は図１５に示すメッセージ本体を参照し、高度アクセス制御設定に基づき、高度アクセス制御を行う。

【００６８】

具体的に、制御ＶＭ３０は、ＶＭ１から送信された割り込みに含まれる図１４のようなメッセージを読み込み、メッセージに指定されている実際の割り込みの宛先であるＶＭ２のＶＭ＿ＩＤを取り出す。次に、制御ＶＭ３０はメッセージに指定されているメッセージ本体を取り出す。

【００６９】

そして、制御ＶＭ３０は取り出した図１５のメッセージ本体を参照し、図１６の高度アクセス制御設定に基づき、ＶＭ１からＶＭ２への割り込みが許可されているかをチェックする。図１５及び図１６は、ＶＭ１からＶＭ２への割り込みが許可されている例を表している。ここでは比較的簡単な高度アクセス制御を制御ＶＭ３０が行う例を示したが、より複雑な高度アクセス制御を制御ＶＭ３０が行うことも可能である。

【００７０】

ＶＭ１からＶＭ２への割り込みが許可されていれば、制御ＶＭ３０はステップＳ１１４に進み、図１４のメッセージ本体を読み込み、読み込んだメッセージ本体を割り込みに含めるメッセージとし、ハイパーバイザ２０に実際の割り込みの宛先であるＶＭ２への割り込みを依頼する。

【００７１】

一方、ＶＭ１からＶＭ２への割り込みが許可されていなければ、不正な割り込み送信依頼であるため、制御ＶＭ３０はステップＳ１１５に進み、ＶＭ１にエラーを返して異常終了する。

【００７２】

ステップＳ１１４に続いてステップＳ１１６に進み、ハイパーバイザ２０は、制御ＶＭ３０からＶＭ２への割り込み送信依頼を受信し、単純アクセス制御設定テーブル２４に保持されている単純アクセス制御設定に基づいて、制御ＶＭ３０からＶＭ２への割り込みが許可されているかをチェックする。制御ＶＭ３０からＶＭ２への割り込みが許可されていれば、ハイパーバイザ２０はステップＳ１１７に進み、制御ＶＭ３０からの割り込みをＶＭ２に送信する。

【００７３】

一方、制御ＶＭ３０からＶＭ２への割り込みが許可されていなければ、不正な割り込み送信依頼であるため、ハイパーバイザ２０はステップＳ１１８に進み、ＶＭ１にエラーを返して異常終了する。

【００７４】

ステップＳ１１７に続いてステップＳ１１９に進み、ＶＭ２は制御ＶＭ３０からの割り込みをハイパーバイザ２０から受信し、必要な処理を実施する。必要な処理はＶＭ２の本体処理部４２で行われるものとする。本体処理部４２は割り込みに含まれるメッセージを取り出し、メッセージの内容に従って処理を行う。

【００７５】

図１５のメッセージ本体の場合、メッセージの内容は特定のパスのファイルの削除を依頼するものとなる。したがって、ＶＭ２の本体処理部４２は特定のパスのファイルの削除を行う。この際、メッセージの内容によっては、ＶＭ１と共有している共有メモリ４７を経由してデータのやり取りを行う場合も考えられる。ＶＭ２は制御ＶＭ３０からの割り込みをハイパーバイザ２０から受信し、必要な処理を実施したあと、正常終了する。

【００７６】

以上、本実施例によれば、ハイパーバイザ２０が高度アクセス制御の機能を持たない場合でも、制御ＶＭ３０を各ＶＭの間に挿入することにより、個別高度アクセス制御に比べて柔軟な高度アクセス制御を行うことができる。

【００７７】

特に、組み込み分野で使用されるようなハイパーバイザ２０は計算能力やメモリ容量の制限のため、最低限の機能だけが実施され、高度アクセス制御のような機能が実装されないことが考えられる。このため、組み込み分野で使用されるようなハイパーバイザ２０は本実施例により得られる効果が大きい。

【００７８】

また、本実施例はＶＭ間割り込みを制御の対象とするものであり、ＶＭ間の共有メモリを制御の対象とする必要がない。一般的に、ＶＭ間通信ではＶＭ間割り込みを制御すれば必要な高度アクセス制御を実現できる。ＶＭ間の共有メモリを制御の対象とすると、ＶＭ間通信では場合によってメモリのコピーなどが発生し、大きなオーバーヘッドとなる恐れがある。

【００７９】

一方、ＶＭ間割り込みの制御は、割り込みを転送するために割り込みの回数が２倍になるが、ＶＭ間割り込み自体のオーバーヘッドが充分に小さいため、大きなオーバーヘッドにならない。本実施例では、ＶＭ間割り込みを制御の対象とし、ＶＭ間の共有メモリを制御の対象としないことにより、オーバーヘヘッドを少なくし、かつ、必要な高度アクセス制御を行うことができる。

【００８０】

以上のように、本実施例では仮想化環境において、ＶＭの数が増大した場合に複雑になるＶＭ間のアクセス制御を効率化する。本実施例では複数のＶＭでなく一つの制御ＶＭ３０で高度アクセス制御を行う為、高度アクセス制御を柔軟に行うことができる。本実施例では、ハイパーバイザ２０でなく制御ＶＭ３０で高度アクセス制御を行う為、ハイパーバイザ２０が高度アクセス制御の機能を持たない場合であっても高度アクセス制御を容易に行うことができる。本実施例によれば、ＶＭ間の通信を容易且つ柔軟に制御できる。

【００８１】

本発明は、具体的に開示された実施例に限定されるものではなく、特許請求の範囲から逸脱することなく、種々の変形や変更が可能である。なお、特許請求の範囲に記載した制御仮想計算機が制御ＶＭ３０に対応し、仮想計算機がＶＭ１～ｎに対応する。

【符号の説明】

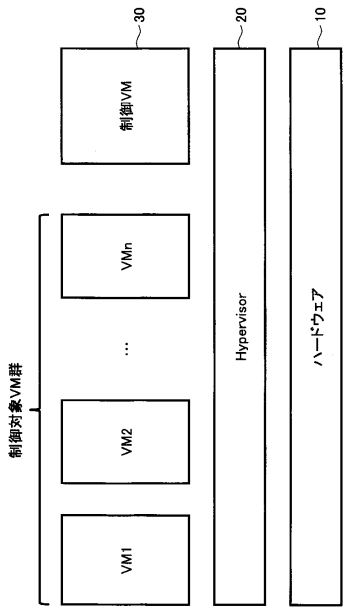
【００８２】

１～ｎ　ＶＭ（一般ＶＭ）
１０　ハードウェア
１１　入力装置
１２　出力装置
１３　ドライブ装置
１４　補助記憶装置
１５　主記憶装置
１６　演算処理装置
１７　インターフェース装置
１８　記録媒体
２０　ハイパーバイザ
２１　割り込み受信部
２２　単純アクセス制御部
２３　割り込み送信部
２４　単純アクセス制御設定テーブル
２５　フック部
３０　制御ＶＭ

３１　割り込み送信部
３２　高度アクセス制御部
３３　割り込み受信部
３４　高度アクセス制御設定テーブル
４１　割り込み送信部
４２　本体処理部
４３　高度アクセス制御部
４４　割り込み受信部
４５　共有メモリ設定テーブル
４６　高度アクセス制御設定テーブル
４７，４８　共有メモリ
５０　高度アクセス制御モジュール
５１　高度アクセス制御部
５２　高度アクセス制御設定テーブル

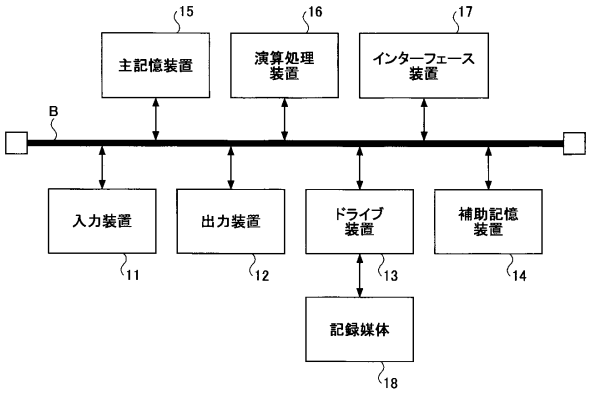
【図１】

本実施例におけるソフトウェアとハードウェアとの関係を表した一例のブロック図



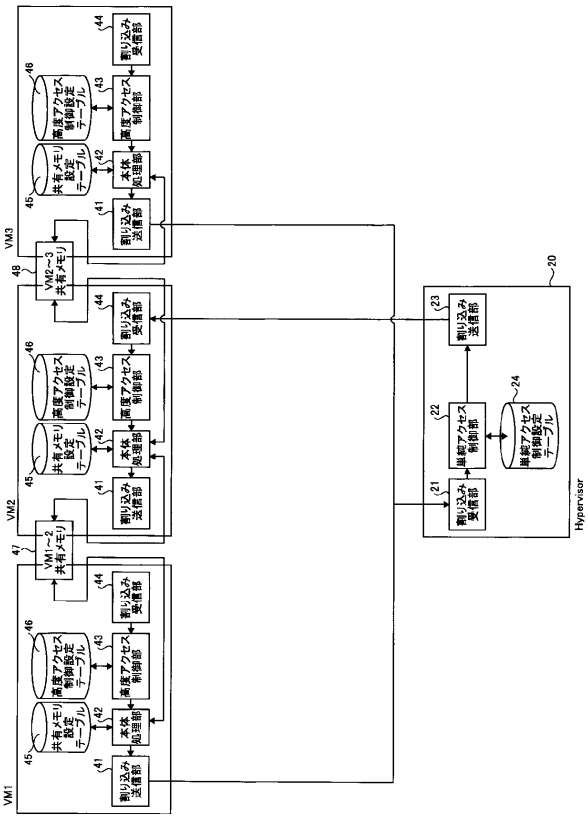
【図２】

本実施例におけるハードウェアの一例の構成図



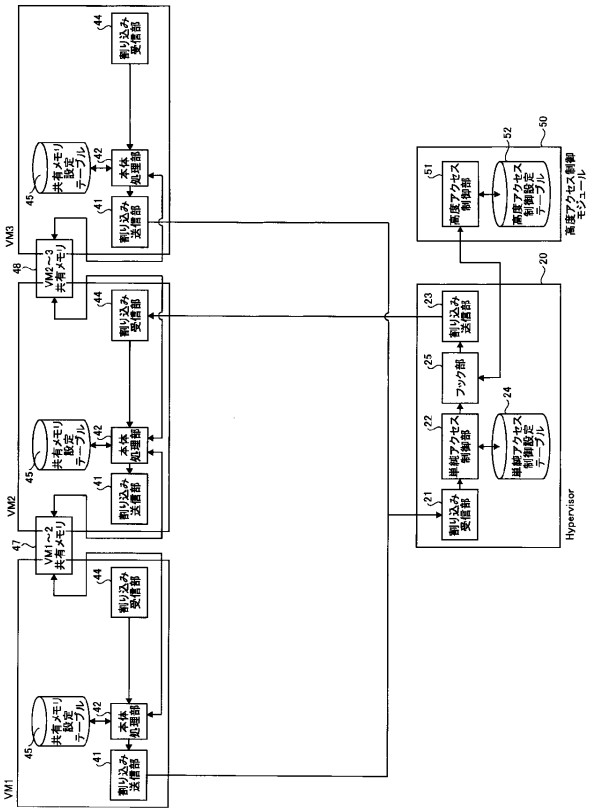
【図３】

個別高度アクセス制御を説明する為の説明図



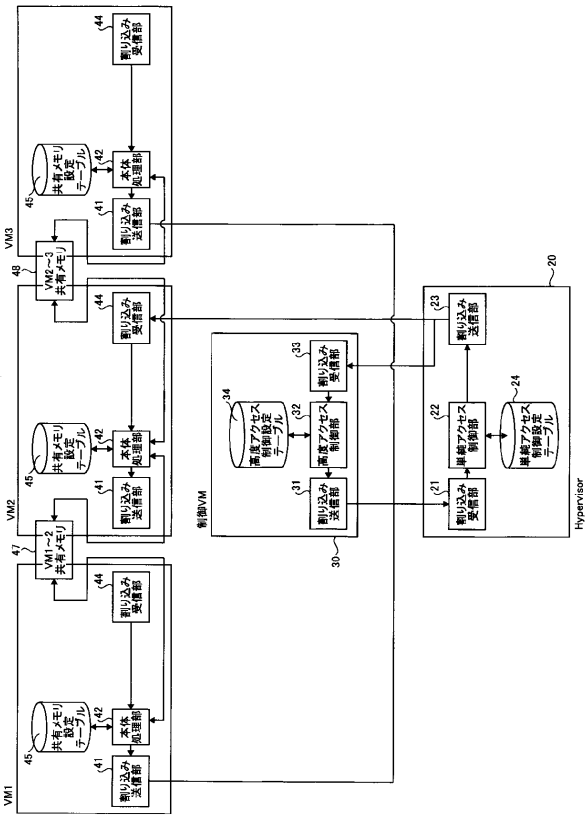
【図４】

ハイパーバイザ高度アクセス制御を説明する為の説明図



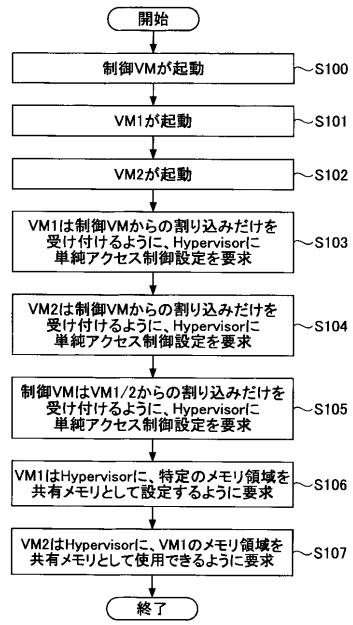
【図５】

制御VMで行う高度アクセス制御を説明する為の説明図



【図６】

準備処理の手順を表した一例のフローチャート



【図７】

単純アクセス制御設定のハイパーコールの例を表した説明図

```
BOOL result = ConfigInterrupt( 0        ,          TRUE);
      結果                  対象のVM ID   割り込みを受け付けるか否か
```

【図８】

単純アクセス制御設定テーブルの一例の構成図

	送信側 VM ID		
受信側 VM ID	0	1	2
0	－	可能	可能
1	可能	－	不可
2	可能	不可	－

【図９】

ハイパーバイザへの共有メモリ設定依頼の例を表した説明図

```
BOOL result = PublishSharedMemory( 0x01                    , TRUE);
      結果                         対象のメモリ領域の番号  共有するかどうか
```

【図１０】

共有メモリ設定テーブルの一例の構成図

VM ID	共有メモリとして設定されたメモリ領域の番号のリスト
1	0x01, 0x05, 0x11, ...

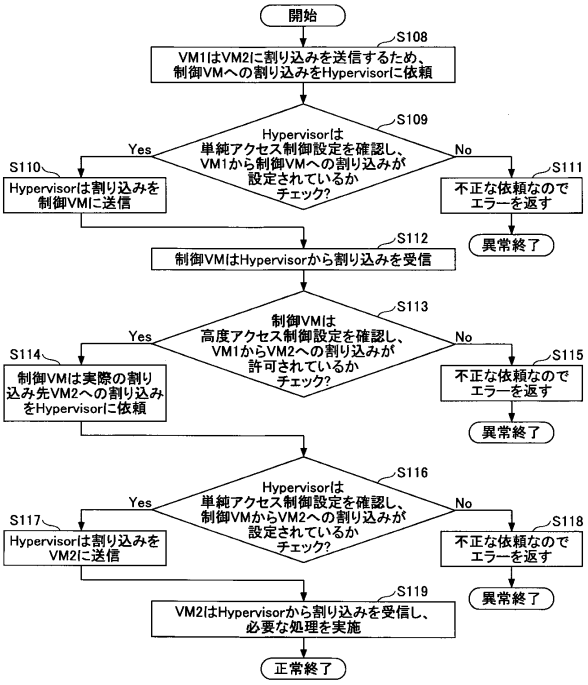
【図１１】

ハイパーバイザへの共有メモリ使用依頼の例を表した説明図

```
BOOL result = UseSharedMemory( 1          , 0x01                    , TRUE);
      結果                     対象のVM ID  対象のメモリ領域の番号  使用するかどうか
```

【図１２】

アクセス処理の手順を表した一例のフローチャート



【図１３】

ハイパーバイザへの割り込み送信依頼の例を表した説明図

```
BOOL result = SendInterrupt( 0x00    , message);
      結果                   VM ID   割り込みに含めるメッセージ
```

【図１４】

割り込みに含めるメッセージの例を表した説明図

実際の宛先のVM ID	割り込みに含めるメッセージ
メッセージ本体	

【図１５】

メッセージ本体に含まれる情報の例を表した説明図

コマンド	rm	←ファイル削除を依頼する
引数1	C:¥public	←対象ファイルのディレクトリ名
引数2	testtext.txt	←対象ファイル名

【図１６】

高度アクセス制御設定の例を表した説明図

宛先VM	コマンド	引数1	引数2
2	rm	"C:¥public"	-

※ 上記に一致するメッセージだけが許可されるものとする

フロントページの続き

審査官　井上　宏一

(56)参考文献　特開２００７－２３３７０４（ＪＰ，Ａ）
特表２００２－５４４６２０（ＪＰ，Ａ）
特開２００７－２１３４６５（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　　９／４６　－９／５４