

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6530723号  
(P6530723)

(45) 発行日 令和1年6月12日(2019.6.12)

(24) 登録日 令和1年5月24日(2019.5.24)

(51) Int.Cl. F I  
**G O 6 F 9/455 (2006.01)** G O 6 F 9/455 1 5 0  
**G O 6 F 9/48 (2006.01)** G O 6 F 9/48 3 0 0 Z

請求項の数 21 外国語出願 (全 19 頁)

(21) 出願番号	特願2016-52123 (P2016-52123)	(73) 特許権者	515348585
(22) 出願日	平成28年3月16日 (2016.3.16)		エーオー カスペルスキー ラボ
(65) 公開番号	特開2016-173821 (P2016-173821A)		AO Kaspersky Lab
(43) 公開日	平成28年9月29日 (2016.9.29)		ロシア国、125212 モスクワ、レニ
審査請求日	平成29年8月17日 (2017.8.17)		ングラドスコ ショス 39エー/3
(31) 優先権主張番号	14/659,049	(74) 代理人	110001139
(32) 優先日	平成27年3月16日 (2015.3.16)		S K特許業務法人
(33) 優先権主張国	米国 (US)	(74) 代理人	100130328
			弁理士 奥野 彰彦
		(74) 代理人	100130672
			弁理士 伊藤 寛之
		(72) 発明者	ヴァチェスラフ アイ. リェヴチェンコ
			ロシア、125212、モスクワ、レニン
			ングラドスコ ショス 39エー/3、カス
			ペルスキー ラボ ザオ シー/オー
			最終頁に続く

(54) 【発明の名称】 コンピュータシステム内における複数のハイパーバイザーの共同運用を容易にするためのシステムおよび方法

(57) 【特許請求の範囲】

【請求項1】

複数のハイパーバイザーの共同オペレーションを調整するためのシステムであって、前記システムは、プロセッサ、データストレージ、及び入力/出力機能、を有するコンピューティング・プラットフォームを備え、

前記プロセッサは、ハイパーバイザー・モードとスーパーバイザー・モードの間の切替が可能であり、

前記ハイパーバイザー・モードは、前記スーパーバイザー・モードよりも高い特権レベルを提供し、

前記コンピューティング・プラットフォームは、命令を含み、

前記命令は、前記コンピューティング・プラットフォームによって実行された時、前記コンピューティング・プラットフォームが、

永続的なハイパーバイザー及び非永続的なハイパーバイザーと、

前記スーパーバイザー・モードでの前記非永続的なハイパーバイザーのオペレーションを調整するように構成されるスケジューラエンジンと、

前記ハイパーバイザー・モードで、前記永続的なハイパーバイザーのオペレーションを調整するように構成されるハンドラ・エンジンと、を実装し、

前記ハンドラ・エンジンは、前記ハイパーバイザー・モードと前記スーパーバイザー・モード間で、前記プロセッサのモード推移の試みを監視しその試みに応答し、

前記ハイパーバイザー・モードから前記スーパーバイザー・モードへのモード推移

の試みに応答して、前記ハンドラ・エンジンは、前記永続的なハイパーバイザーの実行を中断し、及び、前記スーパーバイザー・モードで前記非永続的なハイパーバイザーを実行するようにプロセッサ状態を推移するものであり、

前記スーパーバイザー・モード命令の実行の終了に応答して、前記ハンドラ・エンジンは、前記非永続的なハイパーバイザーの実行を中断し、及び、前記ハイパーバイザー・モードで前記永続的なハイパーバイザーを実行するようにプロセッサ状態を推移するものであり、

前記永続的なハイパーバイザーの実行を中断することと、前記非永続的なハイパーバイザーの実行を中断することには、前記プロセッサの状態を保存することが含まれる、システム。

10

【請求項 2】

前記ハンドラ・エンジンは、オペレーティング・システム・ドライバを使用して実現される、請求項1に記載のシステム。

【請求項 3】

前記ハンドラ・エンジンは、前記ハイパーバイザー・モードと前記スーパーバイザー・モードの間でのプロセッサモード推移を示す前記コンピューティング・プラットフォームの状態情報をモニターするように構成される、請求項1に記載のシステム。

【請求項 4】

前記ハンドラ・エンジンは、前記ハイパーバイザー・モードと前記スーパーバイザー・モードの間でのプロセッサモード推移に関連する少なくとも一つのコマンドをモニターするように構成される、請求項1に記載のシステム。

20

【請求項 5】

前記少なくとも一つのコマンドはvmlrunコマンドを含む、請求項4に記載のシステム。

【請求項 6】

前記ハンドラ・エンジンは、前記ハイパーバイザー・モードと前記スーパーバイザー・モードの間でのプロセッサモード推移に関連する少なくとも一つのパラメータの状態変化を検出するように構成される請求項1に記載のシステム。

【請求項 7】

前記パラメータの状態変化は、ページディレクトリのリロードを含む、請求項6に記載のシステム。

30

【請求項 8】

前記パラメータの状態変化は、CR3レジスタのリロードを含む、請求項6に記載のシステム。

【請求項 9】

前記ハンドラ・エンジンは、インターセプター・エンジンを含み、

前記インターセプター・エンジンは、

前記永続的なハイパーバイザーのオペレーションを制御するための一連のアクションが完了するまで、

モード変更の試みの検出に応答して、

前記プロセッサのモード変更の実行を中断するように構成される、  
請求項1に記載のシステム。

40

【請求項 10】

前記ハンドラ・エンジンは、以下のように構成される、

前記ハイパーバイザー・モードから前記スーパーバイザー・モードへのモード推移の試みに応答して、

前記ハンドラ・エンジンは、前記モード推移の試みがいずれのプロセッサモードにおいて生じたのかを決定し、

前記プロセッサモードの決定に基づいて、

前記永続的なハイパーバイザーの中断は許可され、又は許可されない、

請求項1に記載のシステム。

50

## 【請求項 1 1】

前記ハンドラ・エンジンは、  
 前記非永続的なハイパーバイザーの実行を停止するというコールを示す状況に対応する為  
 に、前記スケジューラエンジンをモニターするように構成され、  
 前記状況に応答して、  
 前記ハンドラ・エンジンは、中断された永続的なハイパーバイザーのオペレーションを再  
 開する、  
 請求項1に記載のシステム。

## 【請求項 1 2】

ハイパーバイザー・モードとスーパーバイザー・モードの間での切り替え可能なプロセ  
 ッサを備えたコンピューティング・システム内における、複数のハイパーバイザーの共同  
オペレーションを調整するための方法であって、

前記ハイパーバイザー・モードは、前記スーパーバイザー・モードより高い特権レベル  
 を提供し、

前記方法は、

前記コンピューティング・システムによって、永続的なハイパーバイザー及び非永続  
 的なハイパーバイザーを実行するステップと、

前記コンピューティング・システムによって、前記ハイパーバイザー・モードで前記  
 永続的なハイパーバイザーのオペレーションを調整するハンドラ・エンジンを実行するス  
 テップを備え、

前記ハンドラ・エンジンを実行するステップにおいて、

前記ハンドラ・エンジンは、前記ハイパーバイザー・モードと前記スーパーバイザ  
 ー・モード間でのプロセッサのモード推移の試みを監視し、その試みに応答し、

前記ハイパーバイザー・モードから前記スーパーバイザー・モードへのモード推移  
 の試みに応答して、前記ハンドラ・エンジンは、前記永続的なハイパーバイザーの実行を  
 中断し、前記スーパーバイザー・モードで前記非永続的なハイパーバイザーを実行するよ  
 うにプロセッサ状態を推移するものであり、

前記非永続的なハイパーバイザーは、スケジューラエンジンによって制御された動作方  
 式に応じて、前記スーパーバイザー・モードで実行され、

前記スーパーバイザー・モードの命令の実行の終了に応答して、前記ハンドラ・エンジ  
 ンが、前記非永続的なハイパーバイザーの実行を中断し、前記ハイパーバイザー・モード  
 で前記永続的なハイパーバイザーを実行するようにプロセッサ状態を推移するものであり

、前記永続的なハイパーバイザーの実行を中断することと、前記非永続的なハイパーバイ  
 ザーの実行を中断することには、前記プロセッサの状態を保存することが含まれる、方法

## 【請求項 1 3】

前記ハンドラ・エンジンは、前記ハイパーバイザー・モードと前記スーパーバイザー・モ  
 ードとの間でプロセッサモード推移を示すコンピューティング・プラットフォームの状態  
 情報をモニターする、請求項12に記載の方法。

## 【請求項 1 4】

前記ハンドラ・エンジンは、前記ハイパーバイザー・モードと前記スーパーバイザー・モ  
 ードとの間のプロセッサモード推移に関連する少なくとも一つのコマンドをモニターする  
 、請求項12に記載の方法。

## 【請求項 1 5】

前記少なくとも一つのコマンドは、vmrunコマンドを含む、請求項14に記載の方法。

## 【請求項 1 6】

前記ハンドラ・エンジンが、前記ハイパーバイザー・モードと前記スーパーバイザー・モ  
 ードとの間のプロセッサモードの推移に関連する少なくとも1つのパラメータの状態変化  
 を検出する、請求項12に記載の方法。

10

20

30

40

50

## 【請求項 17】

前記パラメータの状態変化が、ページディレクトリ又はCR3レジスタのリロードを含む、請求項16に記載の方法。

## 【請求項 18】

前記ハンドラ・エンジンが、  
モード変更を行う試みの検出にตอบสนองして、  
永続的なハイパーバイザーの動作を制御するための一連のアクションが完了するまで、  
プロセッサのモード変更の実行を中断する、  
請求項12に記載の方法。

## 【請求項 19】

前記ハイパーバイザー・モードから前記スーパーバイザー・モードへのモード推移の試みにตอบสนองして、  
前記ハンドラ・エンジンは、前記モード推移の試みがいずれのプロセッサモードにおいて生じたのかを決定し、  
前記プロセッサモードの決定に基づいて、  
前記永続的なハイパーバイザーの中断は許可され、又は、許可されない、  
請求項12に記載の方法。

10

## 【請求項 20】

前記ハンドラ・エンジンは、  
非永続的なハイパーバイザーの実行を停止するというコールを示す状況に対応する為に、  
前記スケジューラエンジンをモニターするように構成され、  
前記状況にตอบสนองして、  
前記ハンドラ・エンジンは、中断された永続的なハイパーバイザーのオペレーションを再開する、  
請求項12に記載の方法。

20

## 【請求項 21】

ハイパーバイザー・モードとスーパーバイザー・モードの間で切り替え可能なプロセッサを備えたコンピューティング・システム内における、複数のハイパーバイザーの共同オペレーションを調整するためのシステムであって、

前記ハイパーバイザー・モードは、前記スーパーバイザー・モードよりも高い特権レベルを提供し、

30

前記システムは、

永続的なハイパーバイザーと非永続的なハイパーバイザーを実行するための手段と、  
前記ハイパーバイザー・モードで前記永続的なハイパーバイザーのオペレーションを調整するようにハンドラ・エンジンを実行するための手段を備え、

前記ハンドラ・エンジンを実行するための手段において、

前記ハンドラ・エンジンは、前記ハイパーバイザー・モードと前記スーパーバイザー・モード間のプロセッサのモード推移の試みを監視し、その試みにตอบสนองし、

前記ハイパーバイザー・モードから前記スーパーバイザー・モードへのモード推移の試みにตอบสนองして、前記ハンドラ・エンジンは、前記永続的なハイパーバイザーの実行を中断し、前記スーパーバイザー・モードで前記非永続的なハイパーバイザーを実行するようにプロセッサ状態を推移するものであり、

40

前記非永続的なハイパーバイザーは、スケジューラエンジンによって制御された動作方式に応じて、前記スーパーバイザー・モードで実行され、

前記スーパーバイザー・モードの命令の実行の終了にตอบสนองして、前記ハンドラ・エンジンが、前記非永続的なハイパーバイザーの実行を中断し、前記ハイパーバイザー・モードで、前記永続的なハイパーバイザーを実行するようにプロセッサ状態を推移するものであり、

前記永続的なハイパーバイザーの実行を中断することと、前記非永続的なハイパーバイザーの実行を中断することには、前記プロセッサの状態を保存することが含まれる、シス

50

テム。

【発明の詳細な説明】

【技術分野】

【0001】

発明分野

【0002】

本発明は、概して情報処理に関するものであり、特に、コンピュータの仮想化技術に関するものである。より詳細には、本発明は、コンピュータシステム内の複数のハイパーバイザーの共同運用に関するものである。

【0003】

背景技術

【0004】

今日のコンピュータシステムは、いろいろな用途（大規模データセンター又はパーソナルコンピューティング）において仮想化技術の利用が増加しているようである。仮想化とは、単一のコンピュータシステム上で、複数のオペレーティング・システムを実行し、利用可能なハードウェアリソースの利用率を最大化するような構成をサポートするものである。

例えば、パーソナルコンピュータのユーザは複数の仮想マシンを利用することができ、それにより、その単一のマシンでの異なるオペレーティング・システムにおいて、様々なアプリケーションを同時に実行することが可能となる。

多くの場合、仮想化は、ウェブホスティングサービスを提供する場合のような大容量サーバにおいて見られる

【0005】

仮想マシンの運用は、一般に、ハイパーバイザーの使用が必要であり、ハイパーバイザーが、仮想マシンの実行を調整し、

仮想マシンのマネージャー又はディスパッチャーとして機能する。一般的に、ハイパーバイザーは、また、仮想マシンモニター（VMM）と呼ばれる。

【0006】

コンピュータシステムに複数のハイパーバイザーがある場合、ハイパーバイザーの運用を適切に調整することが必要となる。仮想マシンにおけるハイパーバイザーの共同運用は、典型的には、それらの設計によって提供される。ハイパーバイザーが、コンピューティング・プラットフォームの制御を行うとき、プロセッサ状態（例えばプロテクションリング）は一時的に変化し、ハイパーバイザーのアクションが完了した後、プロセッサが初期状態に戻される。たとえば、Windowsオペレーティング・システムでは、異なる製造業者による複数の仮想マシンが、同時に動作している場合、スケジューラは、オペレーティング・システム内の通常のスレッドで行う方法と同じ方法で、（例えば、さまざまなスレッドの権限を考慮して）ハイパーバイザーを実行するための時間を割り当てる。

ハイパーバイザーコード自体は、特定のタイムクォンタム時間割当て量の満了時にその実行を正確に完了する。

しかし、このタイムクォンタムは、必ずしもOSのスケジューラによって割り当てられたタイムクォンタムと一致しないことに留意すべきである。

【0007】

複数のハイパーバイザーが同時に使用されるシステムでは、従来のマルチタスク技術が効率的に機能しない可能性があり、最悪の場合には、それらは有効に機能しない場合さえある。これらの問題は、複数のハイパーバイザー間で、他のハイパーバイザーよりも高い特権レベルで実行されるハイパーバイザーが存在する場合、さらに悪化する。

例えば、永続的なハイパーバイザーが、非常に重要な仮想マシンを実行するために使用される場合、1つ又は複数の非永続的なハイパーバイザーと並んで、従来のマルチタスク技術は、永続的なハイパーバイザー用の永続的な運用を適切に行うことを確実にするという点において困難性を有している。もう一つの問題は、他の従来のハイパーバイザーは、永続

10

20

30

40

50

的なハイパーバイザーの存在を認識しないかもしれないし、ハイパーバイザー・モードのプロセッサ上での実行を先に行なおうとしたときに、オペレーティング・システムをクラッシュする可能性があるという問題である。従って、複数のハイパーバイザーの効果的かつ効率的な同時実行を容易にするために、解決策が必要となる。

【0008】

発明の概要

【0009】

本発明の一実施態様は、複数のハイパーバイザーの共同運用を調整するための専用のシステムに対応するものである。システムは、プロセッサ、データストレージ、及び入力/出力機能を有するコンピューティング・プラットフォームを備えている専用機である。プロセッサは、ハイパーバイザー・モードとスーパーバイザー・モードの間の切り替え可能であり、ハイパーバイザー・モードはスーパーバイザー・モードよりも高い特権レベルを提供する。コンピューティング・プラットフォームは命令を含み、その命令を実行すると、コンピューティング・プラットフォームは、永続的なハイパーバイザー及び非永続的なハイパーバイザーと、スーパーバイザー・モードで非永続的なハイパーバイザーの運用を調整するように構成されたスケジューラエンジンと、ハイパーバイザー・モードで、永続的なハイパーバイザーの運用を調整するように構成されたハンドラ・エンジンと、を実装する。

10

【0010】

ハンドラ・エンジンは、ハイパーバイザー・モードとスーパーバイザー・モードの間で、プロセッサのモード推移の試みを監視し、その試みに応答する。ハイパーバイザー・モードからスーパーバイザー・モードへのモード推移の試みに応答して、ハンドラ・エンジンは、プロセッサの状態の保存することを含む、永続的なハイパーバイザーの実行を中断し、スーパーバイザー・モードで非永続的なハイパーバイザーを実行するようにプロセッサ状態を推移する。

20

スーパーバイザー・モード命令の実行の終了に応答して、ハンドラ・エンジンは、プロセッサ状態を保存することを含む、非永続的なハイパーバイザーの実行を中断し、ハイパーバイザー・モードで永続的なハイパーバイザーを実行するようにプロセッサ状態を推移する。

【0011】

【図面の簡単な説明】

【0012】

本発明は、添付の図面に関連し、本発明の様々な実施形態の以下の詳細な説明を考慮して、より完全に理解することができる。

30

【0013】

【図1】図1は、永続的なハイパーバイザー及び1つ又は複数の非永続的なハイパーバイザーを含む、複数のハイパーバイザーが動作している、従来のオペレーティング・システムの一部を示す図である。

【0014】

【図2A】図2Aは、一実施形態に係るシステム内の複数のハイパーバイザーの実行を確実にするためのシステムを示す図である。

40

【0015】

【図2B】図2Bは、一実施形態に係るハンドラ・エンジンの例示的なアーキテクチャーを示すブロック図である。

【0016】

【図3】図3は、一実施形態に係る、図2Aから図2Bに示されたシステムの運用のプロセスを示す。

【0017】

【図4】図4は、本発明の態様の実装がなされた、改善された機能を有する専用機として実行されるコンピュータシステムを示す図である。

50

## 【0018】

本発明は、種々の修正及び代替の形態が容易であり、その具体的態様は、図面に例として示されており、詳細に説明する。しかしながら、その意図は、本発明を記載される特定の実施形態に限定するものではないということが理解されるべきである。それどころか、本意図は、添付の特許請求の範囲によって定義される本発明の精神および範囲内に入るすべての修正、均等物、および代替物をカバーするものである。

## 【0019】

## 【詳細な説明】

## 【0020】

I 用語

10

## 【0021】

以下の用語は、本明細書で使用される用語の定義を規定する。この用語集は本願のみに適用可能である。

## 【0022】

「コンピューティング・プラットフォーム」、「コンピュータ」及び「コンピュータシステム」：相互操作可能な電子デバイスのシステム又は電子デバイスであり、1つ又は複数のプロセッサ、データストレージ、入出力デバイスなどのハードウェアを含む。それは、ハードウェアにより実行されるソフトウェア命令に従って情報を格納及び操作することができる。

それは、1つの物理的なマシンであってもよく、又は、複数の物理的なマシン間に分散してもよい。

20

クラウドコンピューティング分散モデルの場合、役割又は機能によって、又は、プロセススレッドによって、分散することもできる。

例として、ルータ、スイッチなどのネットワークデバイスなどとともに、デスクトップ又はモバイル型のパーソナルコンピュータ(PC)、スマートフォン、及びタブレットを含む。

コンピューティング・プラットフォームは、スタンドアロンデバイス、又は、組み込みデバイスであってもよく、組み込みデバイスは、より大きなデバイス又はシステムの一部であってもよい。

## 【0023】

30

「データストレージ」：1つ又は複数の電子ハードウェアデバイスであって、物理的ストレージ媒体にデータを格納する。例としては、揮発性ストレージ(例えば、ランダムアクセスメモリ(RAM)、静的又は動的)、非揮発性ストレージ(例えば、電気的消去可能プログラマブル読み出し専用メモリ、磁気ディスク、等)が挙げられる。

## 【0024】

「ドライバ」：デバイス(ディスクドライブ等)と、デバイス(オペレーティング・システム・シェル等)を用いるプログラムの間の通訳のような役目を担うエンジン又はコンポーネントである。ドライバは通常、プログラムから汎用コマンドを受け取り、その後、デバイスのために専用のコマンドに変換する。

## 【0025】

40

「エンジン」：実世界のデバイス、コンポーネント、又はコンポーネント配置であり、マイクロプロセッサシステム及びプログラム命令セット(エンジンに特定の機能を実装するようにエンジンを適用させるもの)によって、ハードウェア又はハードウェア及びソフトウェアの組合せを用いて実装される。

それは、(実行される間に)マイクロプロセッサシステムを専用のデバイスに変換する。エンジンはまた、

ハードウェアのみによって容易となる特定の機能、及びハードウェアとソフトウェアの組み合わせによって容易となる他の機能の、両者の組み合わせとして実装することもできる。

ある実装形態では、エンジンの少なくとも一部、場合によっては、全部は、1つ又は複数

50

のコンピュータのプロセッサ上で実行することができ、それは、オペレーティング・システム、システム・プログラム、及びアプリケーションプログラムを実行する。

一方、マルチタスク、マルチスレッド、そこでの適切な分散（例えば、クラスタ、ピア・ツー・ピア、クラウド等）処理、又は他の類似の技術を使用してエンジンを実装する。

さらに、エンジンは、1つ以上のサブエンジンにより、構成されることができ、そのそれぞれが、それ自体でエンジンとみなすことができる。

【0026】

「ハイパーバイザー」：仮想マシンモニター（VMM）とも呼ばれる。それは、プロセッサによって実行可能なプログラムであり、仮想マシンの実行を調整し、仮想マシンのマネージャー又はディスパッチャーとして振舞う。

ハイパーバイザーは、仮想オペレーティングプラットフォームを有するゲスト・オペレーティング・システム又はシステム・プログラムを提示し、且つ、ゲスト・オペレーティング・システム又はシステム・プログラムの実行を管理する。

【0027】

「ハイパーバイザー・モード」：スーパーバイザー・モードよりもさらに高い特権レベルとして関連付けられたプロセッサモードである。ハイパーバイザー・モードは、典型的には、専用の命令を提供する。

その命令により、1つ又は複数のハイパーバイザーの能力を、異なるオペレーティング・システム又はその他のシステム・プログラムに分けることが容易となり、それにより、相互に影響を与えることを回避され、各々がスーパーバイザー・モードで実行される。

典型的に、スーパーバイザー・モードはプロテクションリング0として指定されているシステムでは、ハイパーバイザー・モードは、プロテクションリング1として指定されている。

【0028】

「入力/出力デバイス」又は「入力/出力機能」：コンピュータシステムの電子的ハードウェア部分である。それにより、コンピュータシステムへの入出力する情報の流れを容易にする。

例として、ネットワークインターフェースデバイス、モニター、キーボード、マウス、プリンタ、シリアルポートなどが挙げられる

【0029】

「永続的なハイパーバイザー」：運用の通常の流れで、オペレーティング・システム又は他の不可欠なプログラムのアクティブで継続的なプロセスとして実行されることが期待されているハイパーバイザーである。この永続的な動作方式は、定期的又は非永続的な方式とは異なり、それは、永続的なハイパーバイザーが実行している一方で、定期的なハイパーバイザーのシャットダウンが終了した後に、ハイパーバイザーは特定のタスクの為に時々使用されるという方式である。永続的なハイパーバイザーの例は、非常に重要な仮想マシンの継続的な実行を必要とするハイパーバイザーを含んでおり、又はハイパーバイザーは、ウイルス対策プロセス等として、セキュリティ関連の運用の実行を確実にするために必要である。

【0030】

「プロセッサ」：基本算術演算、論理演算、一時的なストレージ、及び、システムの入力/出力運用を実行することにより、コンピュータプログラムの命令を実行するコンピュータシステムの電子ハードウェア部分である。典型的に、プロセッサは、（単一のチップ上に集積型）マイクロプロセッサとして実装されている。この定義は、複数の相互接続された集積回路上に実装されるプロセッサ回路を含む。現代のプロセッサは、典型的には、複数のプロセッシングコアを含み、複数のプロセッシングコア間で作業負荷を分散することができる。

【0031】

「プロセッサモード」：「特権レベル」とも呼ばれ、あるコンピュータアーキテクチャーのプロセッサ用の動作方式である。それは、運用タイプ及び範囲において、1つ又は複数

10

20

30

40

50



の制限が選択的に適用され、特定のプロセスによって行なわれ、プロセッサによって実行されることができる。例えば、高度に信頼できるカーネルコード、及び、ある場合において、高度に信頼できるドライバ、又はその他のシステム・プログラムは、スーパーバイザー・モードと呼ばれる、無制限（又は比較的少ない制限）モードでの実行が許可されており、一方、他のプロセス（オペレーティング・システムの非監視部を含む）は、ユーザーモードと呼ばれる、比較的より制限されたモードで実行し、且つ、その代わりに制限された運用を実行するために、より信頼できるカーネルを要求するシステムコールを使用する必要がある。プロセッサモードは、プロテクションリングアーキテクチャーをサポートしている。

**【 0 0 3 2 】**

「プロテクションリング」：コンピュータシステムのアーキテクチャー内の権限における、2つ又はそれ以上の階層レベル又はレイヤーである。これは、一般的に、あるアーキテクチャーによって強いられたハードウェアであり、そのアーキテクチャーは、ハードウェア又はマイクロコードレベルで、異なるプロセッサモードを提供する。例えば、リングは、伝統的に、ほとんどの権限（最も信頼されている、通常はゼロが採番される）から、最も少ない権限（最も信頼されてない、通常最も高いリング番号が採番される）に階層的に配置される。典型的には、リング0は、オペレーティング・システムのカーネルが実行されるレベルであり、そして、高い番号のリングは、増大した制限のある特権レベルに関連付けられている。ある現代のプロセッサは、リング0よりもさらに高い特権レベルをサポートしており、それは、リング-1と呼ばれる。

**【 0 0 3 3 】**

「スーパーバイザー・モード」：高度に信頼できるオペレーティング・システムのカーネルプロセス、及び、特定の高度に信頼できるドライバなどの、より高いの特権レベルのプロセスに関連付けられたプロセッサモード。スーパーバイザー・モードは、様々なディスクリプタテーブル用のレジスタを変更するなどの、センシティブなマシンコード運用、又は、割り込みを無効にするなどを行う運用、の実行を許可する。スーパーバイザー・モードは、また、制限されたアドレス空間へのアクセス権を与えることができ、それは、ユーザーモードプロセスによってアクセスできない、メモリ管理ハードウェア、及び、特定の周辺機器である。

**【 0 0 3 4 】**

「システムの仮想マシン」：完全なオペレーティング・システムの実行をサポートする完全なシステムプラットフォームを提供する仮想マシン。これらは通常、既存のアーキテクチャーをエミュレートし、仮想マシンにおける複数のインスタンスを提供するために構築され、それぞれが分けられたコンピューティング環境を提供している。このタイプのアーキテクチャーは、クラウド・コンピューティング・サービス、ウェブホスティング、および他の多くのサービスをサポートしているサーバ上のユビキタスである。システムの仮想マシンは、ホストマシンとして知られている物理的なコンピュータシステムのハードウェア上、又はホストマシンで直接実行するオペレーティング・システムの上で、直接実行できる（オペレーティング・システムレベルの仮想化として知られている）。例としては、Windowsの仮想PC、VMware、Oracle VM等を含む。

**【 0 0 3 5 】**

「ユーザーモード」：アプリケーションプログラム等の低い特権レベルのプロセスに関連付けられたプロセッサモード。ユーザーモードは、データを変更又は破壊、入力/出力機能及び周辺機器にアクセス、様々なプログラムの実行を調整、が可能となるセンシティブな運用の実行を禁止する。ユーザーモードで実行中のプロセスは、その代わりにアクションのこれらの種類を実行するために、より高い特権レベルのプロセスを要求しなければならない。

**【 0 0 3 6 】**

「仮想マシン」は、ホストマシンのハードウェアを使用して、物理マシンのようにプログラムを実行するコンピュータシステムにおけるソフトウェアベースが実装されたものであ

10

20

30

40

50

る。仮想マシンは、実際のマシンのいずれかに対応するその使用及び程度に基づいて、2つの主要な分類（システムの仮想マシン、およびプロセスの仮想マシン）に分けられる。

【0037】

「永続的なハイパーバイザー及び非永続的なハイパーバイザーを実行するための手段」：永続的なハイパーバイザー110a及び非永続的なハイパーバイザー110bを実行するための、1つ又は複数の処理ユニット404、システムメモリ406、及びコード。後述する様々な実施形態の各々のいずれか（又は組合せ）、又は、それらの構造的均等物の各々のいずれかである。

【0038】

「ハンドラ・エンジンを実行するための手段」：ハンドラ・エンジン210を実行するための、1つ又は複数の処理ユニット404、システムメモリ406、及びコード。後述する様々な実施形態の各々のいずれか（又は組合せ）、又は、それらの構造的均等物の各々のいずれかである。

10

【0039】

II. 好ましい実施形態の説明

【0040】

本発明の態様は、他のハイパーバイザーも存在する場合、コンピュータシステム内でのハイパーバイザーの1つの永続的な運用を容易にするコンピュータ的解決に対応するものであり、そこでは、それら他のハイパーバイザーは、同じプロセッサコアで定期的に動作する。この文脈において、ハイパーバイザーの持続性は、ハイパーバイザーが、運用の通常の流れでは、オペレーティング・システム又は他の不可欠なプログラムのアクティブで継続的なプロセスとして実行されることが期待される、ことを意味する。この永続的な動作方式は、定期的又は非永続的な方式とは異なり、それは、永続的なハイパーバイザーが実行している状態の一方で、定期的なハイパーバイザーのシャットダウンが終了した後に、ハイパーバイザーは、特定のタスクの為に時々使用されるという方式である。

20

【0041】

永続的なハイパーバイザー110の例は、非常に重要な仮想マシンの継続的な実行を必要とするハイパーバイザーであってもよく、又は、多くのウイルス対策プロセス等として、セキュリティ関連の運用の実行を確実にするために必要とされるハイパーバイザーであってもよい。非永続的なハイパーバイザー110bの例には、仮想マシンのディスパッチャーとすることができ、また、仮想マシンモニター（VMM）と呼ばれ、それは、California Palo AltoにあるVMWare株式会社製のもの、California Redwood CityにあるOracle株式会社製、及び、Florida Fort LauderdaleにあるCitrix Systems 株式会社によるXen等である。

30

【0042】

永続的なハイパーバイザー110aは、一般的に、非永続的なハイパーバイザー110bのいずれかのものより高い特権レベルで実行されるであろう。例えば、オペレーティング・システムよりも高い特権レベルを有するプロテクションリングをサポートする最新のプロセッサで、ハイパーバイザー110aは、より高い特権レベル（例えば、プロテクションリング1、すなわち、「ハイパーバイザー・モード」）で実行するであろう。一方、非永続的なハイパーバイザー110bは、オペレーティング・システムの特権レベルで実行することになるであろう。

40

【0043】

本発明の態様に係るコンピュータシステムは、様々なエンジンを含み、その各々は、機能又は機能セットが自主的に実行するように、構築、プログラム、構成、又は、それ以外の方法で適用される。本明細書で使用されているエンジンという用語は、実世界のデバイス、コンポーネント、又は、ハードウェアを用いて実装されているコンポーネントの配置を意味している。ハードウェアを用いる場合、例えば、特定用途向け集積回路（ASIC）又はフィールド・プログラマブル・ゲート・アレイ（FPGA）によって、実装される。ハードウェアとソフトウェアとの組み合わせとして、マイクロプロセッサシステム、及び、プログラム命令セット等によって、特別に自主的な機能を実装するようにエンジンに適應する。こ

50

れにより（実行している間に）マイクロプロセッサシステムを専用機に変換される。エンジンは、ハードウェアのみによって容易となる特定の機能、及びハードウェアとソフトウェアの組み合わせによって容易となる他の機能の、両者の組み合わせとして実装することもできる。特定の実施態様では、エンジンの少なくとも一部、及び場合によっては、全てが、1つ又は複数のコンピュータのプロセッサを用いて、実装される。そのコンピュータは、また、エンジンを実装しながら（それにより専用機となる）、オペレーティング・システム、システム・プログラム、及びアプリケーションプログラムを実行する。各エンジンは、物理的及び論理的に適切である様々な構成のいずれかで、物理的に実現することができ、概して、制約が明示的に示されている場合を除き、本明細書に例示的に示される任意で特定の实装に制限されるものではない。また、エンジンは、それ自体が1つ以上のサブエンジンで構成されることができ、その各々は、それ自体でエンジンとみなすことができる。

10

また、本明細書で説明した実施形態で、様々なエンジンの各々は、定義された機能に対応するものである。しかしながら、他の意図される実施形態では、各機能が複数のエンジンに分散されてもよいことが理解されるべきである。同様に、他の意図される実施形態では、複数の定義された機能は、これらの複数の機能を実行する単一のエンジンによって、おそらく他の機能と並んで、又は本明細書に具体的に例に示されているものとは異なって、エンジンセットの中で分散される、ように実装されることができる。

#### 【0044】

図1は、従来のオペレーティング・システム（OS）10の一部を示す図であり、そのOS上で複数のハイパーバイザーが動作している。それは、例えば、永続的に実行されるハイパーバイザー110a、及び、他のハイパーバイザー110b（そこには一つ以上が存在することができる）である。ハイパーバイザー110a及び110bの従来の運用の場合には、従来のタスクスケジューラ120は、各々のハイパーバイザーを実行するための時間を割り当てるだろう。例えば、Windows OSにおいて、スケジューラ120は、優先順位に応じて、マルチタスク制御を管理し、優先度の高いスレッドは第一の優先順位で実行することを意味している。その結果、スケジューラ120を使用しているOS10内で、高い優先順位が割り当てられている場合であったとしても、

20

ハイパーバイザー110の永続的な運用を確保することは不可能である。というも、遅かれ早かれ、スケジューラ120は、ハイパーバイザー110bのコードの実行のために、タイムクォンタムを割り当てるだろう。この文脈では、ハイパーバイザー110bの1つを実行するということは、ハイパーバイザーのもとで動作している仮想マシンの実行であることを本質的に意味している。

30

#### 【0045】

様々な実施形態で、タスクスケジューラ120と同様にハイパーバイザー110a、110bは、各々物理的にエンジンとして実現され、そのエンジンは、コンピューティング・プラットフォームの関連部分（例えば、ハードウェア、必要なファームウェア又はソフトウェアにおける全てのコンポーネントでの組み合わせ）を含んでいる。コンポーネントは、例えば、コンピュータシステムの基本入力/出力システム、デバイスドライバ、ライブラリ、及び他のプログラム命令の関連するパーツであり、プログラム命令は、コンピューティング・プラットフォームのストレージ媒体に格納され、コンピュータシステムのプロセッサ（複数可）に実行されている。

40

#### 【0046】

永続的なハイパーバイザー110aは、いくつかの理由のためにウイルス対策の運用を実行する間に必要となることに留意されたい。

まず、ハイパーバイザーは、OSコアでの悪質なコードの実行（スーパーバイザー・モード、リング0）レベル（例えば、ルートキット）の検出を可能にできる。

次に、ハイパーバイザーは、PatchGuardなどのOSコアプロテクションを避けるために必要とされることができる。

マルチコアプロセッサでは、ハイパーバイザーの個別のコピーは、各コアにおいて動作す

50

るのではなく、異なる対応の前後関係を有している。

【0047】

図2Aは、一実施形態に係るシステム内における複数のハイパーバイザーの実行を確実にするためのシステムを示す。図に示すように、オペレーティング・システム100は、永続的なハイパーバイザー110aを取り仕切る。本実施形態において、永続的なハイパーバイザー110aは、(ハイパーバイザー部110bとは異なり)、スケジューラ120によって制御されおらず、ハンドラ210によって制御され、ハンドラは、システムの状態情報220を監視する。ハンドラ210は、様々な実施形態において、エンジンとして物理的に実装され、そのエンジンは、コンピューティング・ハードウェア及び対応するソフトウェア・コンポーネントを含む。関連する実施形態において、ハンドラ210は、コンピューティング・ハードウェア上で実行される専用のOSドライバを使用して実装されている

10

【0048】

一般的には、プロセッサがハイパーバイザー・モードで(プロテクションリング1)である場合に、任意のハイパーバイザー110bがそのコードの実行を開始しようとする時、プロセッサの動作モードの変更は、最初に生じる。なぜなら、この時間の間に、仮想マシンの次の起動の準備することを担うコマンドが実行されるためである。

例えば、Page Directory及びCR3のレジストリが完全にリロードされる。

仮想マシン110bの起動における他の例としては、(VMWareの場合) vmrunコマンドの実行である。

これらのパラメータとコマンドのいずれかは、システムの状態情報220として、図2Aにまとめて表されている。

20

【0049】

ハンドラ210は、対応するハイパーバイザー110bを実行するかどうかの必要性を見極めるために、特定のパラメータに対する任意の修正又はコマンドの実行(それらを傍受することによって)を監視する。したがって、ハンドラ210は、永続的なハイパーバイザー110aの実行を停止するだろう。

【0050】

図2Bは、一実施形態に係るハンドラ・エンジン210の例示的なアーキテクチャーを示すブロック図である。

コマンド検出エンジン250及びパラメータ変更モニターエンジン252は、システムの状態情報220をモニターするために、各々がプログラムされるか、又は別な方法で構成される。コマンド検出エンジンは、コマンドを検出し、一方、パラメータ変更モニター252は、オペレーティング・システム内の特定のパラメータの変更を検出する。

30

ハイパーバイザー実行検出器256は、検出したコマンドとパラメータの変更を、基準に照らして比較し、ハイパーバイザー110bの実行のためのコールを指示し、又は、永続的なハイパーバイザー110aにおける運用を再開するように適合するためのコールを指示する。このような基準は、プロセッサの状態の変化、Page Directory又はCR3のレジスタのリロード、或いは、vmrunコマンドの生成を含めることができる。

関連する実施形態において、インターセプター・エンジン258は、モード推移の試みにおける、コマンド又はパラメータ変更を示す存在を検出したことに応答して、特定の一連のアクションが行われるまで、そのコマンドの実行又はモード推移を保持するために、

40

プログラムされ、又は別な方法で構成される。

永続的なハイパーバイザー実行制御エンジン260は、以下に詳細に説明するように、そのようなアクションを行うように、プログラムされ、又は別な方法で構成される。

【0051】

以下は、一実施形態に係る、ハンドラ210を構成する他のエンジンと連携して、永続的なハイパーバイザー実行制御エンジン260による、永続的なハイパーバイザー110aのアクティブ化及び非アクティブ化におけるプロセスについての、より詳細な説明である。このプロセスは、次のアクションが含まれる。

50

## 【 0 0 5 2 】

a. プロセッサの以前の状態の保存（例えば、ハイパーバイザー・モードにおいて、ハイパーバイザー110bの1つのコードを実行した最後の瞬間。例：プロテクションリング1）。

## 【 0 0 5 3 】

b. 異なるモードでの永続的なハイパーバイザー110aのコードの実行、例えば、スーパーバイザー・モード（プロテクションリング0）。

## 【 0 0 5 4 】

c. ハイパーバイザー110bにおける任意のコードを実行する試みの決定。

## 【 0 0 5 5 】

d. ハイパーバイザー110bによって、仮想化のアクティブ化のための命令を正確に取り扱うために、プロセッサの以前の状態の復元。

10

## 【 0 0 5 6 】

e. スーパーバイザー・モードでのハイパーバイザー110bのコードを実行するのに先立って、ハイパーバイザー・モードの終了、及び、ハイパーバイザー110aの最後の命令へ制御の推移。

この運用により、

ハイパーバイザー110bがハイパーバイザー・モードで実行され、ハイパーバイザー110bが永続的なハイパーバイザー110aの存在に気づかない場合、永続的なハイパーバイザー110aは、ハイパーバイザー・モードからスーパーバイザー・モードに切り替えられることを確実にする。

20

## 【 0 0 5 7 】

従って、永続的なハイパーバイザー110aのスレッドにおける次の命令は、すでに異なるプロテクションリングの範囲内で、実行が継続されているので、ハイパーバイザー・モード外ですでに実行されている。（一般的に、それはカーネルの優先レベルに対応するリング0になるだろう）

## 【 0 0 5 8 】

図3は、一実施形態に係る運用のプロセスを示す。

工程310において、ハンドラ210は、（コマンド検出器250及びパラメータ変更モニター252を使用して）システムのステータス情報220を監視する。

30

工程320において、ハイパーバイザー実行検出エンジン256が、システムの状態が変化した（例えば、ページディレクトリが上書きされ、又はvmlrunコマンドが実行される）と判断した場合、

その後、工程330で、永続的なハイパーバイザー110aの実行は、永続的なハイパーバイザー実行制御部260によって停止される。

## 【 0 0 5 9 】

関連する実施形態において、工程320では、追加のチェックが、ハイパーバイザー実行検出エンジン256によって実行され、すなわち、CR3レジストリのリロードがいずれのプロセッサモードからコールされたかを決定する。

レジストリのリロードがカーネルモードからコールされた場合、前後関係の単なる切替であると考えられ、

40

そして、ユーザーモードからコールされた場合、例えば、仮想マシンのプロセスからのコールがあったという、さらなる確認をすることが可能である。

このようなチェックにより、システムは、永続的なハイパーバイザー110aの停止の場合に備えて、その停止の必要性が絶対でないのに停止することを防ぐために、「偽のアクティベーション」の数を最小限にすることを可能となる。

## 【 0 0 6 0 】

工程340で、ハイパーバイザー実行検出エンジン256は、永続的なハイパーバイザー110aを再開することについての適性をチェックする。このアクションは、次の条件のうち1つ又

50

は複数に応答することができる。

【0061】

ハイパーバイザー110bの1つを実行するため、スケジューラ120によって割り当てられた実行タイムクォンタムは、満了となった。

【0062】

システム120の状態変化が生じた。

【0063】

ハンドラ210は、ハイパーバイザー110bの実行を監視し、その運用の終了に関する通知を受け取る（例えば、ハイパーバイザーの開発者が提供する、公式アプリケーションプログラマブルインターフェース、又はAPIを介して）。

10

【0064】

工程350で、ハイパーバイザー110bのいずれかが実行を完了したと判定され、その後、工程360において、永続的なハイパーバイザー110aは、永続的なハイパーバイザー実行制御部260により再開される。

複数の異なるハイパーバイザー110bが存在する実施形態では、

最初のハイパーバイザー110bの実行完了後すぐに永続的なハイパーバイザー110aの再開するのに先立ち、

永続的なハイパーバイザー110aは、実行を再開する前に、それらの実行を完了するために、すべてのハイパーバイザー110bを待つ。

【0065】

一実施形態では、永続的なハイパーバイザー110aの運用の再開は、タイマーの手順を用いて行われる。（例えば、Windows OSで、これはKeSetTimerコールを使用して行うことができる）

20

例えば、指定した時間の満了時（実行タイムクォンタムが期限切れとなったとき）に、スケジューラ120は、コードが実行される（例えば、永続的なハイパーバイザー110aのスタート）べきかを決定する。

タイマーは、ハイパーバイザー110bのうちの1つが運用を完了し、永続的なハイパーバイザー110Aを起動することができるように、引き続き起動する次の時間を許可するためにリセットすることができる。

【0066】

永続的なハイパーバイザー110aが停止されるべき時の他の一例は、システム220のステータス情報での、電力関連の変更に関するものである。例えば、「休止状態」モードを実行する時である。

30

この目的のために、電源に関連する機能（WindowsOSではPower Managementのコールバック関数）、又は、システム変数（WindowsOSではSYSTEM\_POWER\_STATE）は、監視されている。

【0067】

永続的なハイパーバイザー110aを停止する必要がある他の例は、一部のコードを起動することである。

それは、また、プロセッサの状態、例えば、エミュレータ（QEMU (Quick Emulator)及び他の類似のプログラム、それらは、ウイルス対策アプリケーションで使用含む）に大きな変化をもたらす。しかし、エミュレータ及び永続的なハイパーバイザー110aが、ウイルス対策製品に含まれている場合、そのから、エミュレータの運用後、後者は自動的に永続的なハイパーバイザー部110aを起動することができる。

40

【0068】

図4は、コンピュータシステム400をより詳細に示す図であり、本明細書に記載されるように、本発明の態様を実装し、改良された機能を有する専用のマシンになる。コンピュータシステム400は、パーソナルコンピュータ402などのコンピューティングデバイスを含んでもよい。パーソナルコンピュータ402は、1つ又は複数の処理ユニット404、システムメモリ406、ビデオ・インターフェース408、出力周辺インターフェース410、ネットワークイ

50

インターフェース412、ユーザ入力インターフェース414、リムーバブル416及び非リムーバブル418メモリアンターフェース、及び、種々のコンポーネントを結合する、システムバス又は高速通信チャネル420、を含む。様々な実施形態では、処理ユニット404は、複数の論理コアを有していてもよく、

論理コアは、システムメモリ406又はメモリ（リムーバブル416及び非リムーバブル418のメモリアンターフェース418に装着された）などのコンピュータ可読媒体上に格納された情報を処理することができる。コンピュータ402のシステムメモリ406は、リードオンリーメモリ（ROM）422等の不揮発性メモリ又は、ランダムアクセスメモリ（RAM）424等の揮発性メモリを含んでもよい。ROM422は、コンピュータ402の他の部分と通信するのに役立つように、基本入力/出力システム（BIOS）426を含むことができる。RAM424は、オペレーティング・システム428、アプリケーションプログラム430、及び、他のプログラム・エンジン432等の、種々のソフトウェアアプリケーションのパーツを格納することができる。さらに、RAM424は、プログラム又はアプリケーションデータ434のような他の情報を格納することができる。様々な実施形態において、RAM424は操作又は動作され、低レイテンシ且つ効率的なアクセスを必要とする情報を格納する。様々な実施形態では、RAM424は、ダブルデータレート（DDR）メモリ、エラー訂正メモリ（ECC）、又は、他のメモリ技術を備え、様々なレイテンシ、及び、RAMBUS 又は DDR2 及びDDR3等の構成を有している。

#### 【0069】

リムーバブル416及び非リムーバブル418のメモリアンターフェースは、コンピュータ402をSSD等のディスクドライブ又は回転ディスクドライブと接続してもよい。

これらのディスクドライブ436は、オペレーティング・システム438、アプリケーションプログラム440、及び、その他のプログラム・エンジン442等の種々のソフトウェアアプリケーションのためのさらなるストレージを提供することができる。さらに、ディスクドライブ436は、プログラム又はアプリケーションデータ444等の他の情報を格納することができる。種々の実施形態では、ディスクドライブ436は、他のストレージ媒体と同様に、低レイテンシを必要としない情報を格納する。

さらに、オペレーティング・システム438、アプリケーションプログラム440のデータ、プログラム・エンジン442、及びプログラム又はアプリケーションデータ444は、上記に示すように、様々な実施形態で、RAM424に格納されているものと同じ情報であってもよく、又は、異なるデータであってもよい。それは、データを格納しているRAM424の潜在的に誘導されるものである。

#### 【0070】

さらに、リムーバブル不揮発性メモリアンターフェース416は、コンピュータ402を、磁気ポータブルディスクに接続し、それは、フロッピーディスク448、Iomega（登録商標）、Zip 又は Jazz、及び光ディスクドライブ450などの磁気媒体を利用するものである。Blu-Ray（登録商標）、DVD-R/ RW、CD-R/ RW及び他の類似のフォーマット等のコンピュータ可読媒体のストレージ用に、光学媒体452を利用する。

さらに他の実施形態では、リムーバブルメモリの容量を増大させるために、ポータブル筐体54内に収容される、SSD又は回転ディスクを利用する。

#### 【0071】

コンピュータ402は、ローカルエリアネットワーク（LAN）458又は広域ネットワーク（WAN）460を介して1つ又は複数のリモートコンピュータ456と通信するためのネットワークインターフェース412を利用してもよい。ネットワークインターフェース412は、通信が可能となるように、ネットワークインターフェースカード（NIC）又は他のインターフェース（モデム462等）を利用することができる。モデム462は、電話回線、同軸ケーブル、光ファイバ、電力線、又は無線経由の通信を有効にすることができる。リモートコンピュータ456は、

類似のハードウェア及びソフトウェア構成を有してもよく、又は、リモートアプリケーションプログラム466を含んでいるメモリ464を有してもよい、そのプログラムは、コンピュータ402に追加のコンピュータ可読命令を提供することができる。様々な実施形態では、

10

20

30

40

50

リモートコンピュータのメモリ464は、識別されたファイル情報などの情報を格納するために利用することができ、ファイル情報は、後でローカルシステムメモリ406にダウンロードすることができる。さらに、様々な実施形態において、リモートコンピュータ456は、アプリケーションサーバ、管理サーバ、クライアントコンピュータ、又はネットワーク機器であってもよい。

#### 【0072】

ユーザは、マウス468及びキーボード470等のユーザ入力インターフェース414に接続された入力デバイスを用いてコンピュータ402に情報を入力することができる。

さらに、入力デバイスは、トラックパッド、指紋スキャナ、ジョイスティック、バーコードスキャナ、メディアスキャナ等であってもよい。

ビデオ・インターフェース408は、モニター472等の表示に視覚情報を提供することができる。

ビデオ・インターフェース408は、組み込みインターフェースであってもよいし、個別のインターフェースであってもよい。

さらに、コンピュータは、コンピュータ402の運用の自由度を増加させるために、複数のビデオ・インターフェース408、ネットワークインターフェース412、及び、リムーバブル416及び非リムーバブル418インターフェースを利用することができる。

さらに、様々な実施形態は、コンピュータ402の性能及び機能を変化させるために、複数のモニター472及び複数のビデオ・インターフェース408を利用する。

他のコンピュータ・インターフェースは、出力周辺インターフェース410等をコンピュータ402に含めることができる。このインターフェースは、コンピュータ402に追加機能を提供するために、プリンタ474又はスピーカ476、或いは、他の周辺装置に接続することができる。

#### 【0073】

コンピュータ402における、様々な代替の構成と実装は、本発明の精神の範囲内にある。これらの変化は、制限はなく、ユニバーサル・シリアル・バス(USB)、プリンターポート、PCIバス、PCIエクスプレス、又は、上記に示した様々なコンポーネントの統合したもの(ノースブリッジやサウスブリッジなどのチップセットコンポーネント)等のシステムバス420に接続された追加のインターフェース、を含んでも良い。

例えば、種々の実施形態において、処理ユニット404は、システムメモリ406からのデータのより効率的(システムバス420が提供するよりも)な転送を可能にするために、組込み型メモリコントローラ(図示せず)を含んでもよい。

#### 【0074】

上記の実施形態は、例示であり、限定を意図するものではない。追加の実施形態は、特許請求の範囲内である。加えて、本発明の態様について、特定の実施形態を参照して説明してきたが、特許請求の範囲によって定義されるような、本発明の範囲から逸脱することなく、変更がなされ得ることを当業者は、認識するだろう。

#### 【0075】

関連技術の当業者は、本発明が上記の任意の個々の実施形態に示したよりも、より少ない機能を備えることができることを認識するだろう。

本明細書に記載の実施形態は、本発明の様々な特徴を組み合わせた方法の網羅的な提示であることを意味するものではない。

従って、実施形態は、明示的な記載がない限り、相互に機能の排他的な組合せではない。むしろ、本発明は、当業者によって理解されるように、異なる個々の実施形態から選択された、異なる個々の機能の組み合わせを含むことができる。

#### 【0076】

上記文書のいずれの援用も、本明細書の明示的な開示に反しない主題の組み込みに限られる。

更に、上記文書のいずれの援用も、

文書に含まれる特許請求の範囲が、本出願の特許請求の範囲に援用されないように制限さ

10

20

30

40

50



れている。

しかしながら、文書のいずれかにおける特許請求の範囲は、具体的に除外されない限り、本明細書の開示の一部として組み込まれる。

上記文書のいずれの援用も、文書で提供される任意の定義が、本明細書に明示的に含まれていない限り、本明細書に参考として援用されないように制限されている。

【 0 0 7 7 】

本発明のための請求項を解釈する目的のために、

米国特許法112条第6項の条文は、具体的な用語“ means for ” 又は “ step for ” が、請求項に記載されてない限り、適用されるものではない、ということが明白に意図されている。

【 図 1 】

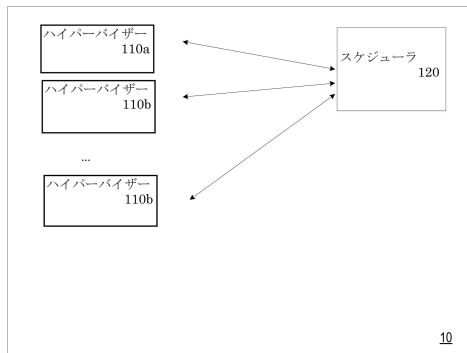


FIG. 1  
先行技術

【 図 2 A 】

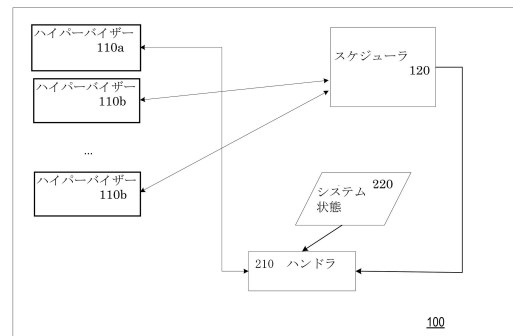


FIG. 2A

【 図 2 B 】

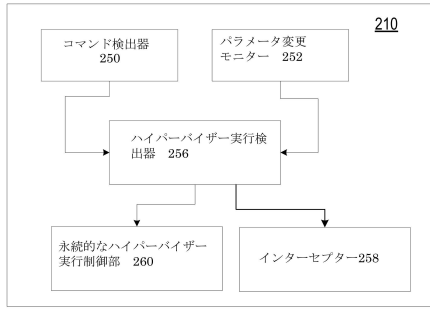


FIG. 2B

【 図 3 】

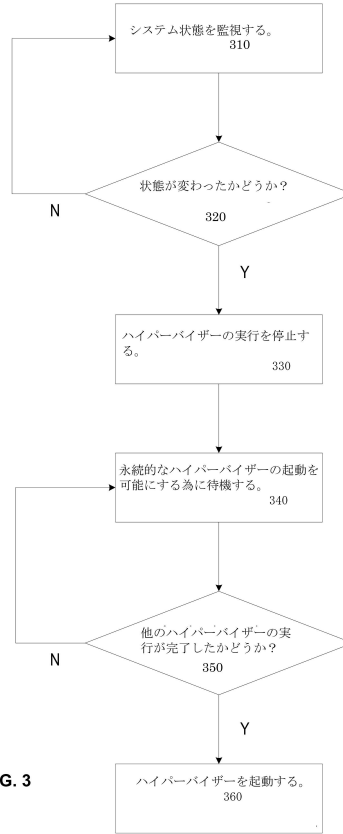


FIG. 3

【 図 4 】

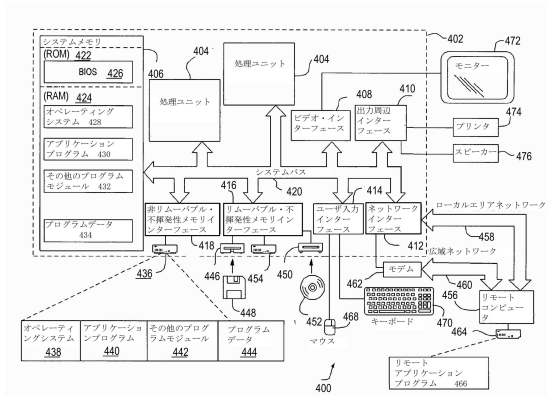


FIG. 4

---

フロントページの続き

(72)発明者 イゴール ワイ . クマギン  
ロシア、125212、モスクワ、レニングラドスコ ショス 39エー / 3、カスペルスキー  
ラボ ザオ シー / オー

審査官 清木 泰

(56)参考文献 特開2011-070663(JP, A)  
特開2009-003749(JP, A)  
米国特許出願公開第2013/0145363(US, A1)  
米国特許出願公開第2011/0066786(US, A1)

(58)調査した分野(Int.Cl., DB名)  
G06F 9/455 - 9/54