

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4690988号
(P4690988)

(45) 発行日 平成23年6月1日(2011.6.1)

(24) 登録日 平成23年2月25日(2011.2.25)

(51) Int.Cl.

F 1

G06F 9/50 (2006.01)
G06F 9/48 (2006.01)G06F 9/46 465Z
G06F 9/46 452H

請求項の数 45 外国語出願 (全 22 頁)

(21) 出願番号 特願2006-266590 (P2006-266590)
 (22) 出願日 平成18年9月29日 (2006.9.29)
 (65) 公開番号 特開2007-102781 (P2007-102781A)
 (43) 公開日 平成19年4月19日 (2007.4.19)
 審査請求日 平成18年9月29日 (2006.9.29)
 (31) 優先権主張番号 11/239,475
 (32) 優先日 平成17年9月30日 (2005.9.30)
 (33) 優先権主張国 米国(US)

(73) 特許権者 591003943
 インテル・コーポレーション
 アメリカ合衆国 95052 カリフォルニア州・サンタクララ・ミッション カレッジ ブールバード・2200
 (74) 代理人 100104156
 弁理士 龍華 明裕
 (72) 発明者 ガウタム チンヤ
 アメリカ合衆国、97123 オレゴン州
 、ヒルズバロ、エス イー 49ティーエイチ シーティー、125
 (72) 発明者 ホン ワン
 アメリカ合衆国、95052 カリフォルニア州、サンタ クララ、ミッション カレッジ ブールバード 2200
 最終頁に続く

(54) 【発明の名称】持続的なユーザレベルスレッド用の装置、システムおよび方法

(57) 【特許請求の範囲】

【請求項 1】

オペレーティングシステム(OS)によりスケジューリングされOS可視シーケンサで実行中のOSスレッドに基づき、命令セットエクステンションを用いて、前記OSによりスケジューリングされるスレッドに対するコンテキストスイッチ動作と独立してOS分離シーケンサで実行されるユーザレベルスレッドである持続的なユーザレベルスレッドを生成する段階を備え、

前記OSは、前記持続的なユーザレベルスレッドの命令に対してスケジューリングを行わず、

前記持続的なユーザレベルスレッドは、前記OSによりスケジューリングされるスレッドがコンテキストスイッチされた時でも前記OS分離シーケンサで動作を継続し、

前記持続的なユーザレベルスレッドは、前記OS可視シーケンサで実行される別のユーザレベルスレッドから生成される方法。

【請求項 2】

前記持続的なユーザレベルスレッドを前記OS分離シーケンサで実行する段階をさらに備える請求項1に記載の方法。

【請求項 3】

前記OSによりスケジューリングされるスレッドおよび前記持続的なユーザレベルスレ

ツドは仮想アドレススペースを共有する
請求項 1 または 2 に記載の方法。

【請求項 4】

前記持続的なユーザレベルスレッドの実行中に OS による処理を必要とするイベントが発生した場合に、前記持続的なユーザレベルスレッドの実行を一時中断する段階をさらに備える請求項 1 から 3 のいずれかに記載の方法。

【請求項 5】

別の OS 可視シーケンサで実行されるサービススレッドに、前記一時中断された持続的なユーザレベルスレッドに対して、OS サービスを提供させる段階をさらに備える請求項 4 に記載の方法。

10

【請求項 6】

前記サービススレッドおよび前記持続的なユーザレベルスレッドは、同じアプリケーションのコンテキストで実行される
請求項 5 に記載の方法。

【請求項 7】

前記持続的なユーザレベルスレッドに対して OS サービスを提供する段階は、
前記別の OS 可視シーケンサに対して、制御移行命令を伝える段階と、
前記制御移行命令を特権レベルで実行する段階と
を有する請求項 5 または 6 に記載の方法。

【請求項 8】

前記特権レベルには Ring 0 レベルが含まれる
請求項 7 に記載の方法。

20

【請求項 9】

前記持続的なユーザレベルスレッドに対して OS サービスを提供する段階は、
前記サービススレッドに前記持続的なユーザレベルスレッドの状態を取得させる段階と
、
前記持続的なユーザレベルスレッドに対して代理実行を行う段階と、
前記持続的なユーザレベルスレッドに実行後状態を戻す段階と
を有する請求項 5 から 8 のいずれかに記載の方法。

【請求項 10】

前記実行後状態から前記持続的なユーザレベルスレッドの実行を再開する段階をさらに有する請求項 9 に記載の方法。

30

【請求項 11】

前記サービススレッドは、前記持続的なユーザレベルスレッドと共有している仮想アドレス空間を通じて前記持続的なユーザレベルスレッドの状態を取得する
請求項 9 または 10 に記載の方法。

【請求項 12】

前記持続的なユーザレベルスレッドは、当該ユーザレベルスレッドを生成したスレッドである前記 OS によりスケジューリングされるスレッドに対するコンテキストスイッチ動作と独立して、動作を継続する
請求項 1 から 11 のいずれかに記載の方法。

40

【請求項 13】

前記命令セットエクステンションは、前記 OS 可視シーケンサおよび前記 OS 分離シーケンサを識別するオペランドを持つ命令を含む
請求項 1 から 12 のいずれかに記載の方法。

【請求項 14】

OS によりスケジューリングされ OS 可視シーケンサで実行中のスレッドに基づき、命令セットエクステンションを用いて、前記 OS によりスケジューリングされるスレッドに対するコンテキストスイッチ動作とは独立して OS 分離シーケンサで実行される ユーザレベルスレッド である持続的なユーザレベルスレッドを生成するシーケンサマネージャ

50

を備え、

前記OSは、前記持続的なユーザレベルスレッドの命令に対してスケジューリングを行わず、

前記持続的なユーザレベルスレッドは、前記OSによりスケジューリングされるスレッドがコンテキストスイッチされた時でも前記OS分離シーケンサで動作を継続し、

前記シーケンサマネージャは、前記OS可視シーケンサで実行される別のユーザレベルスレッドから前記持続的なユーザレベルスレッドを生成する装置。

【請求項15】

前記OSによりスケジューリングされるスレッドに対するコンテキストスイッチ動作とは独立して前記持続的なユーザレベルスレッドを実行する前記OS分離シーケンサをさらに備える請求項14に記載の装置。 10

【請求項16】

前記OSによりスケジューリングされるスレッドおよび前記持続的なユーザレベルスレッドは仮想アドレススペースを共有する

請求項14または15に記載の装置。

【請求項17】

前記シーケンサマネージャは、前記持続的なユーザレベルスレッドの実行中にOSによる処理を必要とするイベントが発生した場合に、前記持続的なユーザレベルスレッドの実行を一時中断する 20

請求項14から16のいずれかに記載の装置。

【請求項18】

前記シーケンサマネージャは、別のOS可視シーケンサで実行されるサービススレッドに、前記持続的なユーザレベルスレッドに対してOSサービスを提供させる

請求項14から17のいずれかに記載の装置。

【請求項19】

前記サービススレッドと前記持続的なユーザレベルスレッドは、同じアプリケーションのコンテキストで実行される

請求項18に記載の装置。

【請求項20】

前記シーケンサマネージャは、前記別のOS可視シーケンサに対して、制御移行命令を伝え、前記制御移行命令を特権レベルで実行させる

請求項18または19に記載の装置。 30

【請求項21】

前記シーケンサマネージャは、前記サービススレッドに前記持続的なユーザレベルスレッドの状態を取得させ、前記持続的なユーザレベルスレッドに対して代理実行を行い、前記持続的なユーザレベルスレッドに実行後状態を戻す

請求項18から20のいずれかに記載の装置。

【請求項22】

前記実行後状態から前記持続的なユーザレベルスレッドの実行を再開する 40

請求項21に記載の装置。

【請求項23】

前記サービススレッドは、前記持続的なユーザレベルスレッドと共有している仮想アドレス空間を通じて前記持続的なユーザレベルスレッドの状態を取得する

請求項21または22に記載の装置。

【請求項24】

前記OS分離シーケンサは、前記持続的なユーザレベルスレッドを生成したスレッドである前記OSによりスケジューリングされるスレッドに対するコンテキストスイッチ動作と独立して、前記持続的なユーザレベルスレッドの動作を継続させる

請求項14から23のいずれかに記載の装置。 50

【請求項 25】

前記命令セットエクステンションは、前記OS可視シーケンサおよび前記OS分離シーケンサを識別するオペランドを持つ命令を含む
請求項14から24のいずれかに記載の装置。

【請求項 26】

システムであって、

OS、少なくとも1つのユーザレベルプログラム、前記プログラムを接続するためのアプリケーションプログラムインターフェース、および前記プログラムの実行をスケジューリングするスケジューラを格納しているメモリと、

少なくとも1つのOS可視シーケンサおよび少なくとも1つのOS分離シーケンサを有するマルチシーケンサと、

OSによりスケジューリングされOS可視シーケンサで実行中のスレッドに基づき、命令セットエクステンションを用いて、前記OSによりスケジューリングされるスレッドに対するコンテキストスイッチ動作とは独立してOS分離シーケンサで実行されるユーザレベルスレッドである持続的なユーザレベルスレッドを生成するシーケンサマネージャとを備え、

前記OSは、前記持続的なユーザレベルスレッドの命令に対してスケジューリングを行わず、

前記持続的なユーザレベルスレッドは、前記OSによりスケジューリングされるスレッドがコンテキストスイッチされた時でも前記OS分離シーケンサで動作を継続し、

前記シーケンサマネージャは、前記OS可視シーケンサで実行される別のユーザレベルスレッドから前記持続的なユーザレベルスレッドを生成する
システム。

【請求項 27】

前記OSによりスケジューリングされるスレッドに対するコンテキストスイッチ動作とは独立して前記持続的なユーザレベルスレッドを実行する前記OS分離シーケンサをさらに備える請求項26に記載のシステム。

【請求項 28】

前記OSによりスケジューリングされるスレッドと前記持続的なユーザレベルスレッドは仮想アドレススペースを共有する

請求項26または27に記載のシステム。

【請求項 29】

前記シーケンサマネージャは、前記持続的なユーザレベルスレッドの実行中にOSによる処理を必要とするイベントが発生した場合、前記持続的なユーザレベルスレッドの実行を一時中断する

請求項26から28のいずれかに記載のシステム。

【請求項 30】

前記シーケンサマネージャは、前記OS可視シーケンサで実行されるサービススレッドに、前記持続的なユーザレベルスレッドに対して、OSサービスを提供させる
請求項26から29のいずれかに記載のシステム。

【請求項 31】

前記シーケンサマネージャは、前記OS可視シーケンサに対して、制御移行命令を伝え、前記OS可視シーケンサに前記制御移行命令を特権レベルで実行させる
請求項30に記載のシステム。

【請求項 32】

前記シーケンサマネージャは、前記サービススレッドに前記持続的なユーザレベルスレッドの状態を取得させ、前記持続的なユーザレベルスレッドに対して代理実行を行い、前記持続的なユーザレベルスレッドに実行後状態を戻す

請求項30または31に記載のシステム。

【請求項 33】

10

20

30

40

50

前記実行後状態から前記持続的なユーザレベルスレッドの実行を再開する
請求項32に記載のシステム。

【請求項34】

前記サービススレッドは、前記持続的なユーザレベルスレッドと共有している仮想アドレス空間を通じて前記持続的なユーザレベルスレッドの状態を取得する
請求項32または33に記載のシステム。

【請求項35】

前記OS分離シーケンサは、前記持続的なユーザレベルスレッドユーザレベルスレッドを生成したスレッドである前記OSによりスケジューリングされるスレッドに対するコンテキストスイッチ動作と独立して、前記持続的なユーザレベルスレッドの動作を継続させる

請求項26から34のいずれかに記載のシステム。

【請求項36】

前記命令セットエクステンションは、前記OS可視シーケンサおよび前記OS分離シーケンサを識別するオペランドを持つ命令を含む
請求項26から35のいずれかに記載のシステム。

【請求項37】

コンピュータに、
OSによりスケジューリングされOS可視シーケンサで実行中のスレッドに基づき、命令セットエクステンションを用いて、前記OSによりスケジューリングされるスレッドに対するコンテキストスイッチ動作とは独立してOS分離シーケンサで実行されるユーザレベルスレッドである持続的なユーザレベルスレッドを生成する手順を実行させ、

前記OSは、前記持続的なユーザレベルスレッドの命令に対してスケジューリングを行わず、

前記持続的なユーザレベルスレッドは、前記OSによりスケジューリングされるスレッドがコンテキストスイッチされた時でも前記OS分離シーケンサで動作を継続し、

前記持続的なユーザレベルスレッドを生成する手順は、前記OS可視シーケンサで実行される別のユーザレベルスレッドから前記持続的なユーザレベルスレッドを生成する手順を有する

プログラム。

【請求項38】

前記コンピュータに、
前記持続的なユーザレベルスレッドを前記OS分離シーケンサで実行する手順をさらに実行させる請求項37に記載のプログラム。

【請求項39】

前記コンピュータに、
前記OSによりスケジューリングされるスレッドおよび前記持続的なユーザレベルスレッドに仮想アドレススペースを共有させる手順をさらに実行させる請求項37または38に記載のプログラム。

【請求項40】

前記コンピュータに、
前記持続的なユーザレベルスレッドの実行中にOSによる処理を必要とするイベントが発生した場合に、前記持続的なユーザレベルスレッドの実行を一時中断させる手順をさらに実行させる請求項37から39のいずれかに記載のプログラム。

【請求項41】

前記コンピュータに、
別のOS可視シーケンサで実行されるサービススレッドに、前記持続的なユーザレベルスレッドに対して、OSサービスを提供させる手順をさらに実行させる請求項37から40のいずれかに記載のプログラム。

10

20

30

40

50

【請求項 4 2】

前記 O S サービスを提供させる手順は、

前記別の O S 可視シーケンサに対して、制御移行命令を伝え、前記制御移行命令を特権レベルで実行させる手順

を有する請求項 4 1 に記載のプログラム。

【請求項 4 3】

前記 O S サービスを提供させる手順は、

前記サービススレッドに、前記持続的なユーザレベルスレッドと共有している仮想アドレス空間を通じて前記持続的なユーザレベルスレッドの状態を取得させる手順

を有する請求項 4 1 または 4 2 に記載のプログラム。

10

【請求項 4 4】

前記 O S 分離シーケンサに、前記持続的なユーザレベルスレッドユーザレベルスレッドを生成したスレッドである前記 O S によりスケジューリングされるスレッドに対するコンテキストスイッチ動作と独立して、前記持続的なユーザレベルスレッドの動作を継続させる

請求項 3 7 から 4 3 のいずれかに記載のプログラム。

【請求項 4 5】

前記命令セットエクステンションは、前記 O S 可視シーケンサおよび前記 O S 分離シーケンサを識別するオペランドを持つ命令を含む

請求項 3 7 から 4 4 のいずれかに記載のプログラム。

20

【発明の詳細な説明】

【背景技術】

【0 0 0 1】

マルチシーケンサコンピュータシステムまたはマルチシーケンサコンピュータプラットフォームにおいて、技術上ならびにコスト上の制約から、特定用途向けのコンピューティング資源、プロセッサまたはプロセッシングユニットの編成が非対称になってしまうことがある。本出願の以下の部分においては、プロセッサまたはプロセッシングユニットも「シーケンサ」または「プロセッシングコア」と呼ぶ。一般的に言って、シーケンサが数多く実装されたコンピュータプラットフォームでは、オペレーティングシステム (O S) がうまく機能しないことがある。特に、複数のシーケンサが非対称に編成された状態で効率よく処理が行えないことがある。

30

【0 0 0 2】

コンピュータプラットフォームは、実装されるシーケンサの種類に関して、その編成が非対称になることがある。例えば、 O S に対して「可視」で、 O S カーネルが管理するシーケンサがある。本明細書では、このようなシーケンサを「 O S 可視シーケンサ」と呼ぶ。ここで、 O S カーネルは、 O S 可視シーケンサの特権状態を管理することができ、 O S 可視シーケンサによってスレッドが実行されている間に、割り込みおよび / または例外を処理する例外ハンドラを提供することができるとしてもよい。このような「 O S 可視シーケンサ」に対し、 O S に対して「不可視」なシーケンサがある。本明細書では、このようなシーケンサを「 O S 分離シーケンサ」と呼ぶ。 O S 分離シーケンサは、 O S コードを実行せず、シーケンサマネージャに管理されるとしてもよい。

40

【0 0 0 3】

最近になって、マルチシーケンサ・ M I M D (複数命令 / 複数データ) 方式 I S A (命令セットアーキテクチャ) エクステンションが提案された。これは、 1 つの O S スレッドコンテキストにおいて複数のユーザレベルスレッドを同時に実行すべく、 O S 分離シーケンサを R i n g 3 のユーザレベルアーキテクチャリソースとして扱うことを可能にする命令セットを定義するものである。このため、 O S 分離シーケンサをユーザレベルアプリケーションに利用できるようになる。ユーザレベル・マルチシーケンサ・ M I M D ・ I S A エクステンションを用いる場合、 O S 可視シーケンサで実行中のアプリケーションスレッドに、 O S から分離されたシーケンサを使用することができるようになる。マルチシ-

50

ケンサ・M I M D・I S Aエクステンションによれば、O Sに対して可視または不可視なシーケンサに処理を任せることにより、アプリケーションは、O Sとは無関係に、多くのシーケンサに対して対応できるようになる。O Sはこの間、O Sに対して可視なシーケンサ上で動作を継続してもよい。このようなユーザレベル・M I M D・I S Aエクステンションによって、ユーザレベルスレッドをO S分離シーケンサ上で実行できるようになり、該ユーザレベルスレッドは、それらを生成し且つO S可視シーケンサで実行されるO Sスレッドと同時に実行されるとしてもよい。

【図面の簡単な説明】

【0 0 0 4】

以下では、添付の図面を用いて本発明の実施形態を詳細に説明し、本発明の内容をより明瞭化にする。 10

【0 0 0 5】

【図1】本発明の一実施形態に係るマルチシーケンサシステムを示すブロック図である。

【0 0 0 6】

【図2】本発明の一実施形態に係るマルチシーケンサシステムの一部であるハードウェアを示す簡略化論理図である。 15

【0 0 0 7】

【図3】本発明の一実施形態に係る、持続的なユーザレベルスレッドのサービスを提供することができるマルチシーケンサシステムを示す簡略図である。 20

【0 0 0 8】

【図4】本発明の一実施形態に係る、持続的なユーザレベルスレッドのサービスを提供することができるマルチシーケンサシステムを示す概念図である。 25

【0 0 0 9】

【図5】本発明の一実施形態に係る、マルチシーケンサシステムの命令セットアーキテクチャを示す図である。 30

【0 0 1 0】

【図6】本発明の一実施形態に係る制御移行命令の命令フォーマットの例を示す簡略図である。 35

【0 0 1 1】

【図7】本発明の一実施形態に係る、監視命令の命令フォーマットの例を示す簡略図である。 40

【0 0 1 2】

【図8】本発明の一実施形態に係る、持続的なユーザレベルスレッドの実行について説明するための簡略化フローチャートである。 45

【0 0 1 3】

【図9】本発明の一実施形態に係る、持続的なユーザレベルスレッドの実行について説明するための簡略化フローチャートである。 50

【0 0 1 4】

説明を簡潔且つ明瞭なものとできるよう、図中に示す構成要素は実寸を反映したものではない。例えば、図示した構成要素の中には分かりやすいようほかの構成要素との相対的なサイズが誇張されているものもある。 55

【発明を実施するための最良の形態】

【0 0 1 5】

以下において、具体的且つ詳細な事例を数多く挙げて本発明の実施形態を細部にいたるまで説明する。しかし、以下で説明する内容以外の実施形態を本発明が含むことは、当業者には明瞭である。また、公知の方法および処理は、本発明の実施形態の説明をあいまいにしない程度に、詳細に説明する。 60

【0 0 1 6】

以下の説明では一部、コンピュータメモリ内でデータビットまたはバイナリデジタル信号に対して行われる処理のアルゴリズムやそのような処理を表す符号を用いている。デー 65

タ処理の分野においては、当業者同士間で作業の内容を伝える場合に、このようなアルゴリズムによる説明を用いる。

【0017】

本明細書内および一般的にアルゴリズムとは、所望の結果を得るために一貫した順序で実施される一連の動作または処理と考えられている。これには、物理量を物理的に操作することも含まれる。必ずしもそうではないが多くの場合、そのような物理量とは電気または磁気信号で、記憶、移動、合成、比較といった操作を行うことができる。こういった信号は、利便性を考慮し一般的な用語を使うべく、ビット、値、成分、符号、文字、項、数字などと呼ばれる。しかし、こういった用語はすべて適切な物理量に対応するものであって、対応する物理量に用いられる便利な名称にすぎない。

10

【0018】

本明細書において、特に明示されていない限り、「処理」「演算」「計算」「決定」等の用語を用いている部分は、コンピュータや演算処理システム、または同様の演算電子デバイスによる処理および/または動作に関する。こういったコンピュータシステムは、システム内のレジスタおよび/またはメモリ内の物理量（例えば電子量）として表されたデータを操作および/または変換して、同システム内のメモリ、レジスタまたは同様の情報記憶/送信/表示装置内に同様に物理量として表される別のデータを得る。

【0019】

本発明の実施形態は例えば、命令または命令セットを格納した機械可読媒体または機械可読装置を用いて実施されるとしてもよい。このような命令または命令セットが機械によって実行されると当該機械は、本発明の実施形態に係る方法および/または動作を行う。このような機械の例を挙げると、適切なプロセッシングプラットフォーム、コンピューティングプラットフォーム、コンピューティングデバイス、プロセッシングデバイス、コンピューティングシステム、プロセッシングシステム、コンピュータまたはプロセッサなどがある。また、このような機械は、ハードウェアおよび/またはソフトウェアを適切に組み合わせることによって実施するとしてもよい。上述の機械可読媒体または機械可読装置は例えば、適切な種類のメモリユニット、メモリデバイス、メモリ装置、メモリ媒体、ストレージデバイス、ストレージ装置、ストレージ媒体および/またはストレージユニットであり、例として、メモリ、取り外し可能/不可能なメディア、消去可能/消去不可能なメディア、書き込み可能/書き換え可能なメディア、デジタル/アナログメディア、ハードディスク、フロッピーディスク、CD-ROM (Compact Disk Read Only Memory)、CD-R (Compact Disk Recordable)、CD-RW (Compact Disk Rewritable)、光ディスク、磁気メディア、さまざまな種類のDVD (Digital Versatile Disk)、テープまたはカセットなどが挙げられる。命令は適切な種類のコードであってもよく、例として、ソースコード、ターゲットコード、コンパイラ型コード、インタープリータ型コード、実行可能コード、スタティックコードまたはダイナミックコードなどが挙げられる。このような命令を実行するには、適切な高レベル、低レベル、オブジェクト指向、ビジュアル、コンパイラ型および/またはインターパリータ型のプログラミング言語を用いるとしてもよい。このようなプログラミング言語の例として、C、C++、Java、BASIC、Pascal、Fortran、Cobol、アセンブリ言語、マシンコードなどが挙げられる。

20

30

Onl y Memory)、CD-R (Compact Disk Recordable)、CD-RW (Compact Disk Rewritable)、光ディスク、磁気メディア、さまざまな種類のDVD (Digital Versatile Disk)、テープまたはカセットなどが挙げられる。命令は適切な種類のコードであってもよく、例として、ソースコード、ターゲットコード、コンパイラ型コード、インターパリータ型コード、実行可能コード、スタティックコードまたはダイナミックコードなどが挙げられる。このような命令を実行するには、適切な高レベル、低レベル、オブジェクト指向、ビジュアル、コンパイラ型および/またはインターパリータ型のプログラミング言語を用いるとしてもよい。このようなプログラミング言語の例として、C、C++、Java、BASIC、Pascal、Fortran、Cobol、アセンブリ言語、マシンコードなどが挙げられる。

40

【0020】

本発明の実施形態は、本明細書に記載する動作を行う装置を含むとしてもよい。このような装置は、所望の目的を達成するように構成されてもよいし、コンピュータ内に格納されたコンピュータプログラムが選択的にアクティブ状態にしたり再構成したりする汎用コンピュータであってもよい。このようなコンピュータプログラムは、コンピュータ可読ストレージ媒体に格納されるとしてもよい。このようなストレージ媒体としてディスクが挙げられ、例えばフロッピーディスク、光ディスク、CD-ROM、光磁気ディスク、ROM (Read Only Memory)、RAM (Random Access Me

50

emory)、E PROM (Electrically Programmable ROM)、EEPROM (Electrically Erasable and Programmable ROM)、磁気/光カードが含まれる。または、ストレージ媒体の例として、その他の種類の、命令を電子状態で格納することに適しておりコンピュータシステムバスに接続が可能な媒体が挙げられる。

【0021】

本明細書で言及する処理およびディスプレイは必ずしも、特定のコンピュータや装置に関するものではない。本明細書の教示に基づいたプログラムとともに用いるのはさまざまな汎用システムであってよいし、所望の方法を実施するためより特化した装置を構築する方が良い場合もあると考えられる。これらさまざまなシステムの望ましい構造は、以下の説明で明らかにする。また、本発明の実施形態の説明は特定のプログラミング言語に基づくものではない。本明細書で説明する本発明の教示は、さまざまなプログラミング言語を用いて実施することができるとしてもよい。

10

【0022】

以下の部分において、図や表、フローチャート、モデルなどさまざまな手段に基づき、本出願で提案される本発明の実施形態を効率よく説明する。当業者には明らかであるが、以下の説明で挙げる実施形態は単なる例にすぎず、本発明を限定するものではない。

【0023】

図1は、本発明の一実施形態に係るマルチシーケンサシステム100を示すブロック図である。

20

【0024】

図1に示すように、マルチシーケンサシステム100は、メモリ102およびマルチシーケンサハードウェア104を備えるとしてもよい。メモリ102は例えば、ユーザレベルプログラム106、スケジューラ108、API (Application Program Interface) 110およびオペレーティングシステム(OS)112を有するとしてもよい。マルチシーケンサハードウェア104は例えば、シーケンサマネージャ114および複数のシーケンサ(例えば、シーケンサ116、118、120および122、図1ではそれぞれSID0、SID1、SID2およびSID3と表す)を有するとしてもよい。図1に図示したシーケンサは4つであるが、マルチシーケンサハードウェア104が有するシーケンサの数は4つに限定されない。

30

【0025】

シーケンサマネージャ114は、ドライバ、OS(例えばOS112)のエクステンション、これ以外のハードウェア、またはハードウェアとソフトウェアの組み合わせとして実施されるとしてもよい。マルチシーケンサハードウェア104は、シーケンサ116、118、120および/または122の種類に関して、対称であってもよいし非対称であってもよい。シーケンサ116、118、120および/または122のうち幾つかは、自分の実行リソースセットを有する物理プロセッサであってもよい。もしくは、シーケンサ116、118、120および/または122のうち幾つかが、ほかの論理プロセッサと物理リソースを共有する論理プロセッサ(例えばハイパースレッド)であってもよい。シーケンサ116、118、120および122を実現するハードウェアコアは、1つであってもよいし複数の独立したものであってもよい。

40

【0026】

マルチシーケンサシステム100の例を挙げると、デスクトップパソコン、ワークステーション、サーバーコンピュータ、ラップトップコンピュータ、ノート型コンピュータ、携帯型コンピュータ、携帯情報端末(PDA)、携帯電話、ゲームコンソールなどがあるが、これらに限定されない。

【0027】

メモリ102の例を挙げると、半導体デバイス、光学デバイス、磁気デバイスのいずれか、もしくはこれらの組み合わせが考えられるが、これらに限定されない。半導体デバイスとしては、SDRAM (Synchronous Dynamic Random A

50

ccess Memory) デバイス、RDRAM (RAMBUS DRAM) デバイス、DDR (Double Data Rate) メモリデバイス、SRAM (Static RAM) デバイス、フラッシュメモリ (FM) デバイス、EEPROM (Electrically Erasable Programmable Read Only Memory) デバイス、不揮発性 (NV) RAM デバイス、取り外し可能なUSB (Universal Serial Bus) メモリデバイスといったものが例として挙げられる。光学デバイスには、CD-ROM (Compact Disk ROM) などがある。磁気デバイスには、ハードディスク、フロッピーディスク、磁気テープなどがある。メモリ 102 は、システム 100 内に固定されるとしてもよいし、システム 100 から取り外し可能な構成としてもよい。

10

【0028】

本発明の一実施形態によれば、ユーザレベルプログラム 106 が API 110 を介してスレッドライブラリにアクセスするとしてもよい。API 110 は、ユーザレベルプログラム 106 に対して、スレッドの作成、制御および同期を行うとしてもよい。スケジューラ 108 は、マルチシーケンサハードウェア 104 上での実行に向けて、スレッドの命令をスケジューリングするとしてもよい。

【0029】

マルチシーケンサハードウェア 104 の例を挙げると、複数のシングルスレッド型 / マルチスレッド型中央演算処理装置 (CPU)、デジタル信号プロセッサ (DSP)、縮小命令セットコンピュータ (RISC)、復号命令セットコンピュータ (CISC) などがあるが、これらに限定されない。また、マルチシーケンサ 104 は特定用途向け集積回路 (ASIC) の一部であってもよいし、特定用途・一般顧客向け LSI (ASSP) の一部であってもよい。

20

【0030】

本発明の一実施形態によると、マルチシーケンサハードウェア 104 は、例えば論理プロセッサであるシーケンサ 116、118、120 および 122 (SID0 ~ SID3) を実装したシングルコアプロセッサであってもよい。シングルコアプロセッサ 104 は、例えば並列マルチスレッディング (CMT)、同時マルチスレッディング (SMT) および / または Switch-on-Event マルチスレッディング (SOEMT) に対応可能としてもよい。一実施形態によれば、シーケンサ SID0 ~ SID3 はそれぞれ独自の次命令ポインタ論理を有し、シングルコアプロセッサ 104 はスレッドの命令をすべて実行するとしてもよい。論理プロセッサ SID1 ~ SID3 はそれぞれ、アーキテクチャ状態の独自のバージョンを保持するが、シングルコアプロセッサ 104 の実行リソースは、複数の並列スレッド、同時スレッドおよび / または Switch-On-Event スレッドの間で共有するとしてもよい。

30

【0031】

上述した一実施形態に係るマルチシーケンサハードウェア 104 はシーケンサ毎に 1 つのスレッドという構成であるとしてもよいが、本明細書の開示はシングルスレッドプロセッサに限定されない。本明細書で説明する技術内容は、CMP (チップマルチプロセッシング) システムまたは SMP (同時マルチスレッディングプロセッサ) システムにも同様に応用することができる。また、CMP プロセッサの各コアは SMP プロセッサまたは Switch-On-Event マルチプロセッサ (SoeMP) である、CMP プロセッサと SMP プロセッサを備えるハイブリッドシステムにも応用できる。例えば、シングルチップハードウェアパッケージ 104 内のマルチスレッドプロセッシングコアを備えるシステムに、本明細書で開示する技術内容を利用するとしてもよい。

40

【0032】

上述したが、シーケンサ SID0 ~ SID3 は、例えば処理速度、処理能力や電力消費量といった演算能力に影響する要因に関して、必ずしも均一である必要はなく非対称であってもよい。例えば、シーケンサ SID0 は、所定の命令セットアーキテクチャ (例えば、32 ビットプロセッサに対応した IA-32 命令セットアーキテクチャ) において命令

50

の大半を処理するように構成されている「主」シーケンサとしてもよい。そして、シーケンサ S I D 1 は、これらの命令のうち選択された命令サブセットを処理する「副」シーケンサであるとしてもよい。本発明の別の実施形態によれば、主プロセッサは副プロセッサよりも早い速度で命令を処理するものと定義してもよい。また、OS に対して可視のシーケンサと不可視のシーケンサがあるとしてもよい。例えば、シーケンサ S I D 0 がOS 1 1 2 に対して可視で、シーケンサ S I D 1 ~ S I D 3 はOS 1 1 2 から分離されているとしてもよい。しかし、主シーケンサがOS に対して可視で、副シーケンサがOS から分離されているというわけではない。OS は、分離状態にあるシーケンサ（このようなシーケンサを本明細書ではOS 分離シーケンサと呼ぶ）上で実行される命令に対してスケジューリングを行わないということである。

10

【0033】

本発明の実施形態によると、マルチシーケンサシステム 1 0 0 は持続的なユーザレベルスレッド（P U L T）のサービスを提供するとしてもよい。P U L T については、以下で図 2 から図 8 を参照しつつ詳述する。本出願において、通常のユーザレベルスレッド（以下では単にユーザレベルスレッドと呼ぶ）は、ユーザレベルまたはRing 3 レベルで実行される非特権命令だけから成る一連の命令であるとしてもよい。ユーザレベルスレッドは通常、OSスレッドに「水面下で」サポートされるとしてもよい。このようなOSスレッドは、ネイティブOSスレッドと呼んでもよいし、単にネイティブスレッドとしてもよい。このOSスレッドのコンテキストにおいて、少なくとも 1 つのユーザレベルスレッドがOS可視シーケンサまたはOS分離シーケンサの一方で実行されるとしてもよい。OS がOSスレッドをコンテキストスイッチすると、このOSスレッドに属するすべてのシーケンサのコンテキストが保存 / 復元される。従って、これらのシーケンサで動作中のユーザレベルスレッドは中断 / 再開される。本発明の一実施形態によれば、持続的なユーザレベルスレッド（P U L T）は、OS分離シーケンサ上で動作するユーザレベルスレッドである。P U L T が実行される環境は例えば、該 P U L T を生成する別のユーザレベルスレッドの仮想アドレススペース（V A S）である。P U L T の実行は、当該 P U L T を生成したユーザレベルスレッドの実行とは無関係に、従って同期せずに行われるとしてもよい。ユーザレベルスレッドが属するOSスレッドがコンテキストスイッチされた時でも、P U L T はOS分離シーケンサ上で動作を継続するとしてよい。本明細書では、このような状態、つまり P U L T の生成元であるスレッドに関するOSスレッドのコンテキストスイッチ動作とは関係なく、ユーザレベルスレッドが動作を継続する特性を「持続的」と呼ぶ。

20

【0034】

図 2 は、本発明の一実施形態に係るマルチシーケンサシステムの一部であるハードウェア 2 0 0 を示す簡略化論理図である。

30

【0035】

ハードウェア 2 0 0 は、図 1 に示すマルチシーケンサシステム 1 0 0 の一部であるマルチシーケンサハードウェア 1 0 4 に対応するとしてもよい。マルチシーケンサハードウェア 1 0 4 は、図 1 に示すように、シーケンサマネージャ 1 1 4 および、例えば複数のシーケンサ 1 1 6、1 1 8、1 2 0 および 1 2 2 を有するとしてもよい。本発明の一実施形態によれば、シーケンサマネージャ 1 1 4 は、シーケンサ 1 1 6、1 1 8、1 2 0 および 1 2 2 がユーザレベルプログラム 1 0 6 に対して均一且つ対称であるように見えるように、シーケンサ 1 1 6、1 1 8、1 2 0 および 1 2 2 を仮想化できるとしてもよい。つまり、図 1 のシーケンサマネージャ 1 1 4 に対応するシーケンサマネージャ 2 0 1 は、シーケンサ 2 1 1、2 1 2、2 1 3 および 2 1 4 の非対称性を隠して、図 2 に同じサイズの正方形で示すように、アセンブリ言語プログラマが論理的に見た場合、シーケンサ 2 1 1、2 1 2、2 1 3 および 2 1 4 が均一且つ対称に見えるようにする。シーケンサ 2 1 1、2 1 2、2 1 3 および 2 1 4 は図 1 に示す論理プロセッサ S I D 0、S I D 1、S I D 2 および S I D 3 に対応するとしてもよい。

40

【0036】

50

図3は、本発明の一実施形態に係る、持続的なユーザレベルスレッドのサービスを提供できるマルチシーケンサシステム300を示した簡略図である。

【0037】

本発明の一実施形態によれば、アプリケーション301（例えば図1に示すユーザレベルプログラム106）の実行はOS302が管理するとしてもよい。OS302は、各々がユーザレベルスレッドを含んでいる可能性がある1以上のネイティブスレッドを生成およびスケジューリングして、アプリケーション301を実行するとしてもよい。このようなネイティブスレッドの例を挙げると、例えばOS302に対して可視である一連のシーケンサ（例えばシーケンサ311および312）のうちの1つで実行される図3のネイティブスレッド303がある。OS302は、OSカーネルに基づきシーケンサ311および312を管理するとしてもよい。OS302はさらに、シーケンサ311および312の特権状態を制御し、シーケンサ311および312で生じる割り込みおよび例外を処理するとしてもよい。

【0038】

本発明の一実施形態によれば、シーケンサマネージャ310は、OSに対して可視であるシーケンサ311および312ならびに、別の一連のシーケンサ、例えばOS302から分離されておりOS302のカーネルが管理していないシーケンサ313、314、315、316、317および318へのアクセスを制御できるとしてもよい。シーケンサ313、314、315、316、317および318は、OS分離シーケンサまたは「OS不可視」シーケンサである。

【0039】

本発明の一実施形態によれば、OSがスケジューリングしたスレッド（例えば、ユーザレベルスレッドを含む可能性があるネイティブスレッド）は、ハードウェア、ソフトウェアもしくは両者の組み合わせによって構成されるインターフェースを介して、シーケンサマネージャ310と通信を行うとしてもよい。OS302に不可視なシーケンサ（例えばシーケンサ313、314、315、316、317および318）を使用したい場合、OSがスケジューリングしたスレッド（例えばネイティブスレッド303）が、OS分離シーケンサのうちの1つ（例えばシーケンサ313）で動作するユーザレベルスレッド（例えばスレッド304）を生成するように、シーケンサマネージャ310に要求を送るとしてもよい。本発明の一実施形態によれば、ユーザレベルスレッド304は、ネイティブスレッド303のOSコンテキストスイッチ動作とは無関係なシーケンサ313で実行されるので、持続的なユーザレベルスレッド（PULT）であるとしてもよい。

【0040】

以下においては、ネイティブスレッド303を例として、持続的なユーザレベルスレッドの作成および実行について説明する。以下の説明がほかのスレッド、例えばネイティブスレッドに属するユーザレベルスレッドにも当てはまることは、当業者には明らかである。

【0041】

本発明の一実施形態によれば、OS分離シーケンサ313は、OS可視シーケンサ311で実行されるユーザレベルスレッド303と共有する仮想アドレススペース（VAS）320で、PULT304を実行するとしてもよい。VAS320は、OS302によってアプリケーション301を実行するために生成されるとしてもよい。本発明の一実施形態によれば、VAS320が共有されているので、シーケンサマネージャ310は、OS可視シーケンサ311で実行されるユーザレベルスレッド303の実行環境、例えば、仮想アドレスマップを取得して、この実行環境をOS分離シーケンサ313で実行されるPULT304に用いることができるとしてもよい。本発明の一実施形態によれば、OS分離シーケンサ313は、仮想アドレススペースの変化を自動的に記録できるスヌープ方式のTLB（変換索引バッファ）を有するとしてもよい。ここで、TLBとはプロセッサ内にある小型のキャッシュで、高速で検索できるように仮想アドレスから物理アドレスへの変換を記録するとしてもよい。プロセッサは、このようなTLBを用いることによって、

10

20

30

40

50

仮想アドレスを物理アドレスへと変換する場合の不必要的ページウォークを避けることができる。

【0042】

図4は、本発明の一実施形態に係る、持続的なユーザレベルスレッドのサービスを提供できるマルチシーケンサシステム400を示す概念図である。

【0043】

本発明の一実施形態に係るマルチシーケンサシステム400は、複数のOS分離シーケンサ（例えばシーケンサ411、412、413および414）およびOS可視シーケンサ（例えばシーケンサ405および406）を管理するシーケンサマネージャ410を備えるとしてもよい。OS402は、ユーザレベルアプリケーション（例えばアプリケーション401Aおよび401B）の実行を管理するとしてもよい。OS402は、1以上のスレッド（例えば、OS可視シーケンサ405および406で動作するユーザレベルスレッド403および404）ならびにネイティブスレッドを生成することによって、アプリケーション401Aおよび401Bを管理する。

【0044】

シーケンサマネージャ410は、OS402から分離されておりOS402が管理していないシーケンサ411、412、413および414へのアクセスに基づき、アプリケーション401Aおよび401Bに対して実行リソースを提供するとしてもよい。なお、この場合のアクセスは、OS可視シーケンサ405および406で実行されているスレッド403および404またはOSがスケジューリングするネイティブスレッドからの要求に基づいて、行われるとしてもよい。この結果、アプリケーション401Aおよび401Bは、ユーザレベルスレッド403および/または404、もしくはネイティブスレッドに基づき、OS分離シーケンサ411、412、413および414にアクセスできるようになるとしてもよい。

【0045】

本発明の一実施形態によると、シーケンサマネージャ410は、ユーザレベルスレッド403および404によってアプリケーション410Aおよび401B用に生成された持続的なユーザレベルスレッド実行のために、さまざまなタイムスロットをシーケンサ411、412、413および414に割り当てるとしてもよい。例えば、図4に示した参照符号t₁、t₂、t₃、t₄、t₅およびt₆はそれぞれ、PULTであるT₁、T₂、T₃、T₄、T₅およびT₆の実行用に割り当てられたタイムスロットを示すものであつてもよい。

【0046】

本発明の一実施形態によると、シーケンサマネージャ410は、OS分離シーケンサ（例えば、シーケンサ411、412、413および414）用に、割り込み記述子テーブル（例えば、テーブル421、422、423および424）を設定するとしてもよい。テーブル421を例に挙げて説明すると、割り込み記述子テーブルはシーケンサ411で発生するイベントを処理するイベントハンドラ431を含むとしてもよい。OS分離シーケンサ（例えば、シーケンサ411）上でPULT（例えばPULT T₃）を実行している間（例えば、t₃）、OS402による対処が必要となるイベントや状況（例えばページフォルトまたはシステムコール）が発生する可能性がある。この場合、イベントハンドラ431はPULT T₃の実行を一時中断し、当該イベントが発生したときのPULT T₃の実行状態を保存するとしてもよい。

【0047】

本発明の別の実施形態によると、ページフォルトやシステムコールが発生した場合、OS不可視シーケンサ（例えばシーケンサ411）は、OS可視シーケンサ（例えば、PULT T₃を起動したシーケンサ405）の実行に直接割り込むとしてもよい。この場合、続いてシーケンサ411は、トリガをシーケンサマネージャ410に送って割り込み処理を要求する。本発明のさらに別の実施形態によると、OS分離シーケンサ411はRing 0サービスチャネル、もしくはページフォルトまたはシステムコールを処理するチ

ヤナルをプログラムするとしてもよい。ここでチャネルとは、アーキテクチャ状態（例えば、トリガがアーキテクチャが定義された一連のプロセッサ状態であり、レスポンスがサービスルーチンであるトリガ・レスポンス・マッピングを含む）を保持するレジスタであってもよい。またチャネルは、例えばユーザレベル命令によってプログラムされるとしてもよい。ページフォルトまたはシステムコールが発生した場合、OS 分離シーケンサ 411 は、そのアドレスがチャネル内で特定されているハンドラを起動するとしてもよい。

【0048】

PULT T₃ の中断は、同じアプリケーション 401A を実行するための新しいユーザレベルスレッド（例えば、スレッド 409）が OS 可視シーケンサで実行されるようスケジューリングされるまで、継続するとしてもよい。スレッド 409（サービススレッドとも呼ぶ）は、一時中断されている PULT があるかどうか、および例えば代理実行によって提供される OS サービスを待っている PULT があるかどうか確認または検証するよう、シーケンサマネージャ 410 に対して要求するとしてもよい。代理実行は、シーケンサマネージャ 410 から PULT T₃ の状態を取得することによって、PULT T₃ を模倣するスレッド 409 に基づいて行われるとしてもよい。PULT T₃ の実行状態は、アドレススペース（例えば、共有仮想アドレススペース（VAS）、図 3 を参照のこと）を共有しているサービススレッド 409 によって再開されるとしてもよい。スレッド 409 は続いて、PULT T₃ を一時停止させた原因である実行条件に対して処理を行うとしてもよい。スレッド 409 はこの処理が終了すると、実行後状態を保存し、当該状態をシーケンサマネージャ 410 に戻すとしてもよい。シーケンサマネージャ 410 は続いて、当該状態を PULT T₃ に渡して、PULT T₃ の実行を再開するとしてもよい。

【0049】

上述した代理実行により、アプリケーションプログラマに対してマルチシーケンサシステムが対称性を有すると見せかけることができ、マルチシーケンサシステム（例えばシステム 100）が非対称性を有するものとして構築されたことを隠すとしてもよい。

【0050】

図 5 は、本発明の一実施形態に係るマルチシーケンサシステムの命令セットアーキテクチャ（ISA）500 を示す図である。

【0051】

ISA は、アセンブリ言語プログラマ、バイナリトランスレータ、アセンブラーなどから見た場合のシステムの論理図を定義するものである。ISA 500 は、論理ストレージ 502 および命令セット 504 を備えるとしてもよい。論理ストレージ 502 は、マルチシーケンサシステム（例えば、図 1 に示したマルチシーケンサシステム 100）用の可視メモリ階層、アドレス割当方法、レジスタセットなどを定義するとしてもよい。一方命令セット 504 は、命令およびマルチシーケンサシステム 100 がサポートする命令のフォーマットを定義するとしてもよい。

【0052】

命令セット 504 は、IA-32 命令セットとして公知である命令セットを含むとしてもよい。本発明の実施形態に係る命令セットは、これらに限定されないが、IA-32 命令セットのエクステンションやそれ以外の命令セットであってもよい。本発明の一実施形態によると、命令セット 504 は例えば、SXFR 制御・移行命令およびそのエクステンションを含む E-SXFR 命令、ならびに SEMONITOR 監視命令およびそのエクステンションを含む E-SEMONITOR 命令を有するとしてもよい。同じ OS ネイティブスレッドに属する 2 つのシーケンサの間で行われ得る SXFR および SEMONITOR とは違い、E-SXFR および E-SEMONITOR は、異なる特権レベルで動作する 2 つの別々の OS ネイティブスレッドにそれぞれ属する 2 つのシーケンサの間で実行することができる。例えば、一方のスレッドは OS コンテキストスイッチされるユーザレベルアプリケーションであり、他方のスレッドは、デバイスドライバのように、特権レベルで持続的に実行される PULT であり、OS コンテキストスイッチには影響されないとし

10

20

30

40

50

てもよい。

【0053】

本発明の一実施形態によると、E-SXFR命令を用いて、第1シーケンサから第2シーケンサに信号を送り、E-SEMONTOR命令を用いて、第1シーケンサから送信される信号を監視するよう、第2シーケンサを構成するとしてもよい。また、E-SXFR制御移行命令およびE-MONITOR監視命令はシーケンサを認識しており、シーケンサを認識したほかの複合命令を構築するために用いられるとしてもよい。

【0054】

本発明の一実施形態によると、E-SXFR命令は図6に例示する命令フォーマットを持つとしてもよい。図6に示すように、E-SXFR命令はOPコード602、オペランド604、606、608、610および612を含むとしてもよい。本発明の一実施形態によると、シーケンサマネージャは、一時中断された持続的なユーザレベルスレッドを処理するための実行環境を作りだすために、シーケンサの特権状態などのオペランドを有するE-SXFR命令を提供するとしてもよい。

10

【0055】

一実施形態によると、オペランド604は、E-SXFR命令信号の送信先であるデスティネーション/ターゲットシーケンサのシーケンサID(SID)に対応するとしてもよい。オペランド606は、シナリオまたは制御メッセージを含むとしてもよい。このようなシナリオまたは制御メッセージは、条件または予測されるイベントを表す、そのアーキテクチャが定義された識別コードであってもよい。シナリオは非同期制御移行に影響を与えるとしてもよい。この詳細は表1を参照して後述する。

20

【0056】

一実施形態によると、オペランド608は、E-SXFR命令を実行するシーケンサで命令を実行するかどうか決定する条件となるパラメータを含むとしてもよい。図6に条件パラメータと示すそのようなパラメータの例を挙げると、「WAIT」パラメータまたは「NO-WAIT」パラメータがある。例えば、E-SXFR命令が代理実行シナリオで用いられる場合、WAIT条件パラメータに基づき、E-SXFR命令を実行するシーケンサでの命令の実行を取りやめ、別のシーケンサでの代理実行の完了を保留にする。一方、NO-WAIT条件パラメータであれば、E-SXFR命令を実行するシーケンサでの命令の実行が別の命令を実行するシーケンサでの代理実行と並行して継続するとしてもよい。

30

【0057】

一実施形態によると、オペランド610は、シナリオ固有ペイロードまたはデータメッセージを含むとしてもよい。例えばFORKシナリオの場合、該ペイロードはオペランド604が示すシーケンサでの実行が始まる箇所である命令ポインタを含むとしてもよい。別の実施形態によると、ペイロードオペランド610は命令ポインタ、スタックポインタ、一連の制御レジスタなどを含むとしてもよい。ペイロードオペランド610に含まれるアドレスは、例えばリテラル、レジスタ間接および/またはベース/オフセットアドレッシングなど、さまざまなアドレス指定モードで指定するとしてもよい。

【0058】

40

一実施形態によると、オペランド612は、オペランド604に含まれるSIDに関するルーティング方法またはルーティング関数を特定するとしてもよい。ルーティング関数は、E-SXFR命令を実行した結果生成される信号が、ブロードキャスト信号、ユニキャスト信号またはマルチキャスト信号のうちどの信号として送信されるかを制御する。ルーティング関数はさらに、例えば、信号を配信するためのルーティングにおいて基本構成要素であるシーケンサ間のインターネットをアシストするために用いられるトポロジー固有情報のような情報を符号化するとしてもよい。

【0059】

一実施形態によると、E-SEMONTOR指示は、図7に例示するような命令フォーマットを有するとしてもよい。図7に示すように、E-SEMONTOR命令はOP

50

コード 702 およびオペランド 704、706 および 708 を含むとしてもよい。オペランド 704、706 および 708 は、持続的なユーザレベルスレッドの実行環境の形成を手助けするべくシーケンサの特権状態に関する情報を含むとしてもよい。例えば、オペランド 704 はシナリオ ID を含み、シナリオを特定するとしてもよい。オペランド 706 は、シーケンサ ID (SID) および命令ポインタ (EIP) に関する情報を含むとしてもよい。説明の便宜上、オペランド 706 に含まれる情報を本明細書では、図 7 に示すように、「SIDEIP」と呼ぶ。

【0060】

E - SEMONITOR 命令は、オペランド 704 で指定されたシナリオとオペランド 706 で指定された SIDEIP をマッピングする。シナリオの SIDEIP へのマッピングを本明細書では、「サービスチャネル」のマッピングと呼ぶとしてもよい。オペランド 708 によってプログラマは所定のサービスチャネルをどのように処理されるか特定または制御するための制御パラメータを入力できるようになるとしてもよい。例えば、プログラマは E - SEMONITOR 命令に基づき、シーケンサが監視するサービスチャネルをプログラムするとしてもよい。

10

【0061】

上記の説明から、E - SXFR 命令および E - SEMONITOR 命令は両方とも、特定のシーケンサを指定しているオペランドを有することから、「シーケンサを認識した」命令であることが分かる。

20

【0062】

図 8 および図 9 は、本発明の実施形態に係る、持続的なユーザレベルスレッドの実行を説明するための簡略化フローチャートである。両図において、同じ参照番号は同じ動作を示している。

【0063】

本発明の一実施形態によると、ブロック 801 に示すように、OS 分離シーケンサ 822 が持続的なユーザレベルスレッドを実行するとしてもよい。本明細書では説明の便宜上、当該持続的なユーザレベルスレッドを、PULT-1 と呼ぶ。PULT-1 は例えば、第 1 OS 可視シーケンサによって実行される第 1 ユーザレベルスレッドによって生成されるとしてもよい。PULT-1 の実行中、OS のサービスが必要なイベントが 1 以上発生するとしてもよい。ここで、以下に詳述するように、第 2 ユーザレベルスレッドが必要な OS サービスを PULT-1 に対して提供するとしてもよい。第 2 ユーザレベルスレッドは PULT-1 を生成した第 1 ユーザレベルスレッドと同じユーザレベルスレッドであってもよいし、違うものであってもよい。また、第 2 OS 可視シーケンサ (例えば、OS 可視シーケンサ 821) またはこれ以外の OS 可視シーケンサで実行するとしてもよい。

30

【0064】

本発明の一実施形態によると、図 8 および図 9 のブロック 802 で示すように、PULT-1 を実行中の時間 t_1 において OS サービスを必要とするイベントが発生するとしてもよい。このイベントを検出すると、ブロック 803 に示すように、シーケンサマネージャ 820 は時間 t_1 (例えば、論理ポインタ) での PULT-1 の状態を保存する手順を開始するとしてもよい。ブロック 803 に示すように PULT-1 の状態が保存されると、ブロック 804 に示すように PULT-1 の実行は時間 t_2 から一時中断されるとしてもよい。本発明の一実施形態によると、OS サービスを必要とするイベントについて通知を受け取った後、外部割込みに基づき、シーケンサマネージャ 820 は PULT-1 の実行を一時中断するとしてもよい。PULT-1 の実行の一時中断は、例えばサービススレッドによる代理実行に基づき必要な OS サービスの提供が行われるまで続くとしてもよい。

40

【0065】

本発明の一実施形態によると、ブロック 805 に示すように、ある時点において、OS がスケジューリングしたスレッド (例えば新規ユーザレベルスレッド) が OS 可視シーケンサ 821 で動作を開始するとしてもよい。この新規ユーザレベルスレッドは、PULT

50

- 1と同じアプリケーション用に実行されるとしてもよい。このスレッドは、例えば代理実行によって、PULT-1に対してOSサービスを提供するとしてもよく、このため本明細書ではサービススレッドと呼んでもよい。OSサービスの提供を開始する前に、ブロック806に示すように、当該サービススレッドは、OSサービスの提供を待つ一時中断されたPULTがあるかどうかシーケンサマネージャ820に確認または検証するとしてもよい。

【0066】

このサービススレッドは、OSサービスの提供を待つスレッドとして、PULT-1を特定するとしてもよい。本発明の一実施形態によると、ブロック807に示すように、続いてサービススレッドは、例えば共有仮想アドレススペース(VASS)に基づきPULT-1の実行状態を取得することによって、PULT-1を模倣、つまり真似するとしてもよい。サービススレッドによって模倣されたPULT-1は、該サービススレッドを実行するOS可視シーケンサで実行され、OSサービスを受け取るとしてもよい。この結果、ブロック808に示すように、サービススレッドの代理実行によってPULT-1に対してOSサービスが提供され、PULT-1が一時中断された条件についてOSによる処理が行われる。本発明の一実施形態によると、OS可視シーケンサ(例えば、OS可視シーケンサ821)で動作するユーザレベルスレッドは、シナリオを用いてアドレス位置を監視することによって、一時中断中のPULTを検出するとしてもよい。ここで、サービスシナリオは、例えばキャッシュミスを監視するための条件である。キャッシュミスが発生してその結果例えばしきい値を超えた場合、サービスチャネルのハンドラが起動されるとしてもよい。

【0067】

本発明の一実施形態によると、PULT-1が一時中断された条件に対してOSが行う処理が完了すれば、ブロック809に示すように、PULT-1の実行後状態が保存され、シーケンサマネージャ820を介してOS分離シーケンサ822に戻される。OS分離シーケンサ822は、サービススレッドから実行後状態を受け取った後、ブロック810に示すようにPULT-1の実行を再開するとしてもよい。

【0068】

表1に、本発明の実施形態に係るシナリオの例をまとめて示す。

【0069】

【表1】

SPU 特性		出方向サービスシナリオ	入方向サービスシナリオ
OSに対する可視性	OS 不可視	OSによる処理を必要とし、Ring間移行が生じる動作。 例： □ ページフォルト処理 □ I/O アクセス	受信制御メッセージ □ Go_shred □ Halt_shred □ Report/Self □ Save/Restore_shred □ Single-step_shred
	OS 可視	送信制御メッセージ □ Fork □ Join □ Yield □ Wait □ Scatter/Set □ Gather/Get □ CSAVE/RSTOR □ Go/Halt/Single-step	OSによる処理を必要とし、Ring間移行が生じる動作。例： □ ページフォルト処理 □ I/O アクセス
コアの複雑性	単純	SSE13 命令	
	複雑		SSE13 実行

【0070】

本発明の実施形態によると、表1に示したシナリオには出方向サービスと入方向サービ

10

20

30

40

50

スの 2 つに関するものがある。各サービスについて、動作および制御メッセージは、S P U (S h r e d P r o c e s s i n g U n i t) の特性の 1 つである、O S に対する可視性に基づき特徴付けるとしてもよい。出方向サービスの中で、R N A (R e s o u r c e n o t a v a i l a b l e : 利用可能なリソースなし) のカテゴリに当てはまる動作のシナリオがある。このR N A というカテゴリは、O S 分離シーケンサで利用不可能なリソースに対するアクセスのため、シーケンサで実行中に生成されたイベント用のものである。本発明の一実施形態によると、R N A のカテゴリに入るシナリオは例えば、直接 O S サービスをアクティブにできないO S 分離シーケンサ上のページフォルト処理および / または I / O アクセスを含むとしてもよい。

【 0 0 7 1 】

10

本発明の実施形態によれば、上述した動作のシナリオには一連の送信制御メッセージが付随する。一実施形態によれば、このようなメッセージは例えば、表 1 の出方向サービスシナリオの欄に示したメッセージを含むとしてもよい。しかし、本発明はこれらに限定されず、示したもの以外のメッセージを用いるとしてもよい。制御メッセージはO S 可視シーケンサに送信される。

【 0 0 7 2 】

上述した出方向サービスシナリオに関連して、O S 分離シーケンサからのサービス要求に対してリソースを提供するO S 可視シーケンサのカテゴリに入る入方向サービスシナリオがある。表 1 の入方向サービスシナリオの欄に示すように、これらのサービスは出方向サービスシナリオにそのまま対応するが、O S 可視シーケンサを介して利用可能である。一実施形態によると、入方向サービスに付随する受信制御メッセージには、表 1 の入方向サービスシナリオの欄に示したメッセージが含まれるとてもよい。しかし、本発明はこれらに限定されず、示したもの以外のメッセージを用いるとしてもよい。受信メッセージはO S 分離シーケンサから受信するとしてもよい。

20

【 0 0 7 3 】

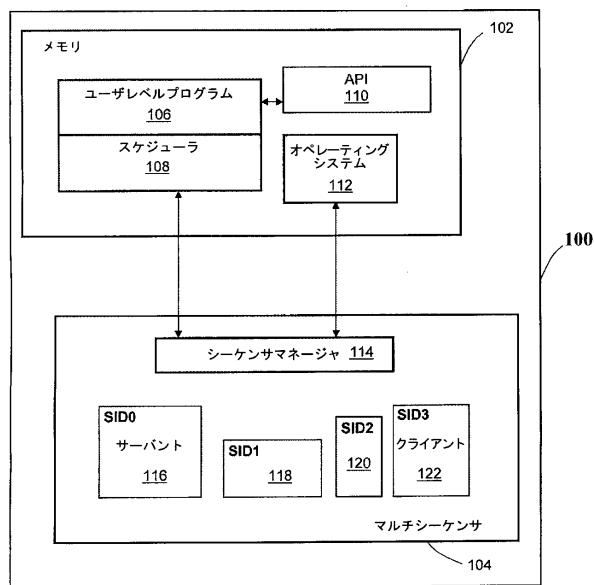
本発明の実施形態によると、出方向サービスに対処する命令は例えば、コアの複雑性という点から見れば比較的単純なS S E 1 3 命令である。また、例えばO S 分離シーケンサからの入方向サービスまたは要求に対処する命令は、概してより複雑なものである。

【 0 0 7 4 】

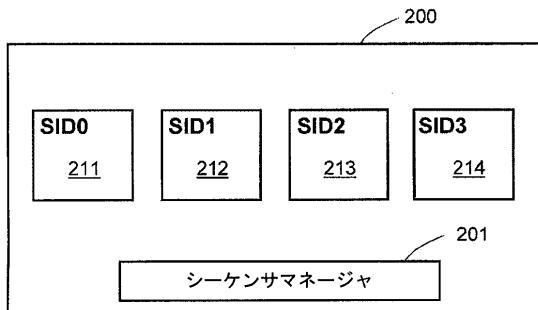
30

本明細書では本発明の特徴を具体的に説明してきたが、当業者であれば数多くの変形例や代替例、変更および同様の技術内容に想到することは明らかである。このため、本願の請求項は、本発明の範囲内にある変形例や変更はすべて含むものと解釈されるべきである。

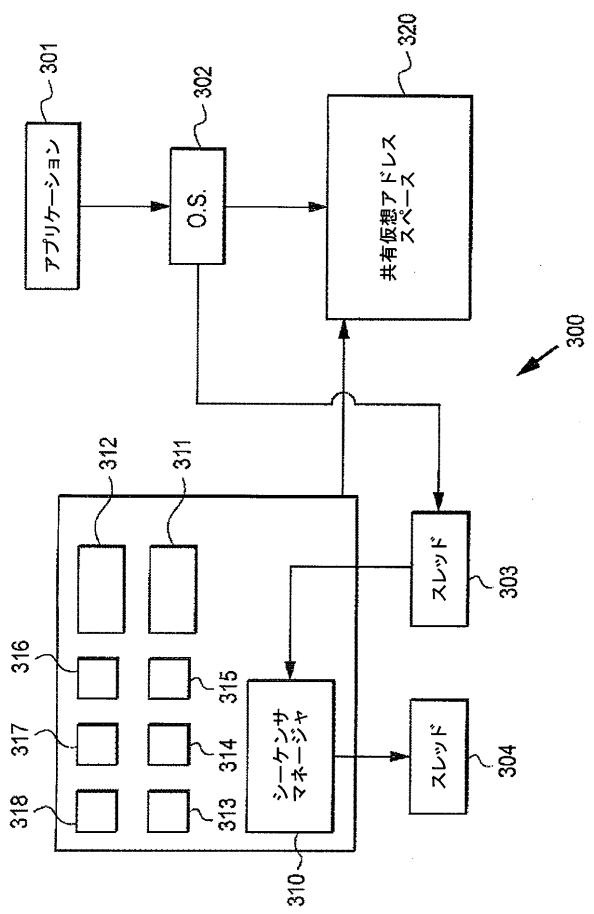
【 図 1 】



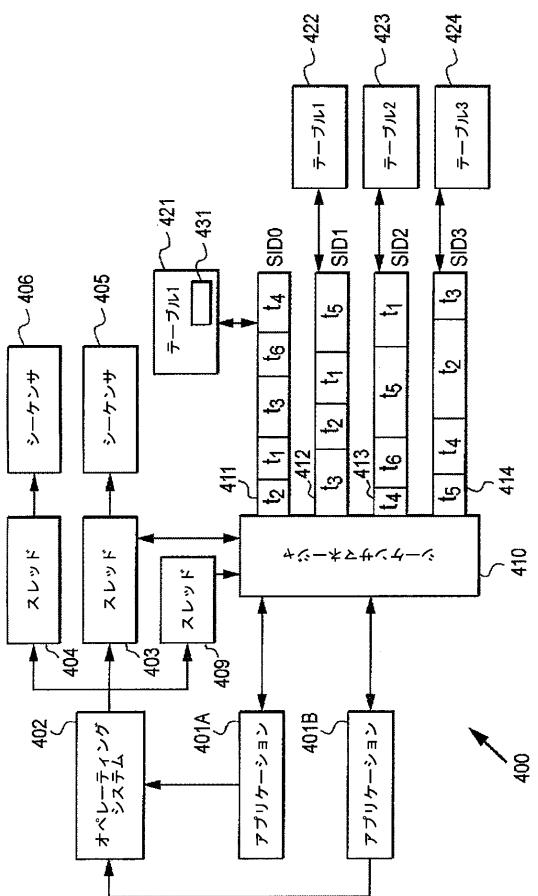
【 図 2 】



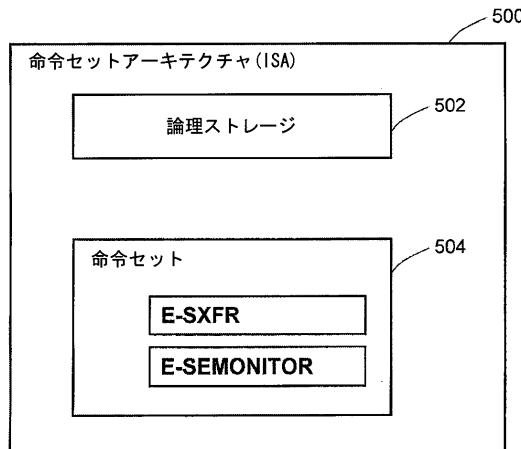
【図3】



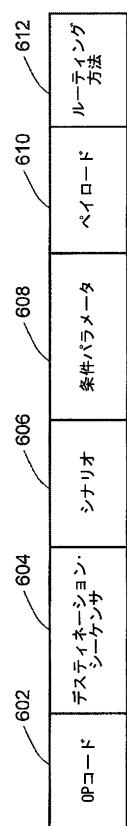
【 四 4 】



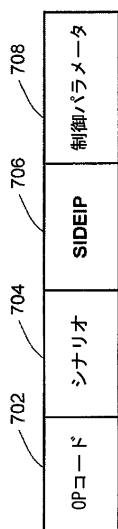
【図5】



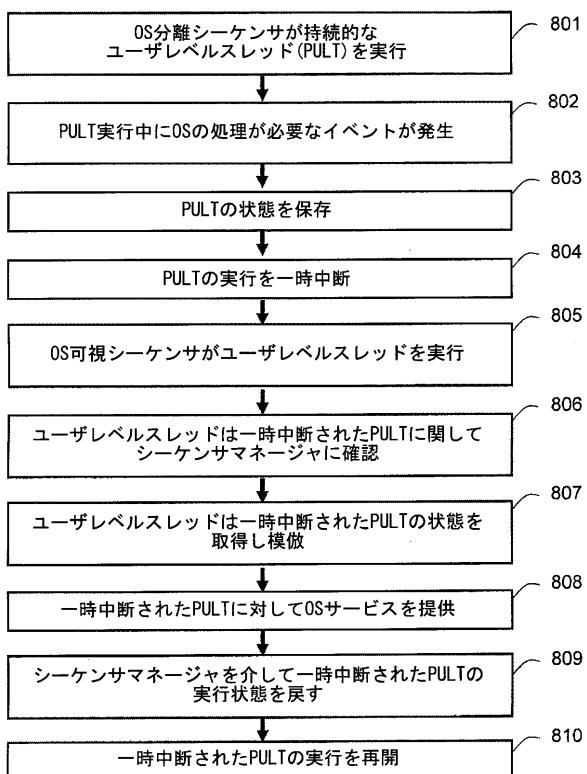
【図6】



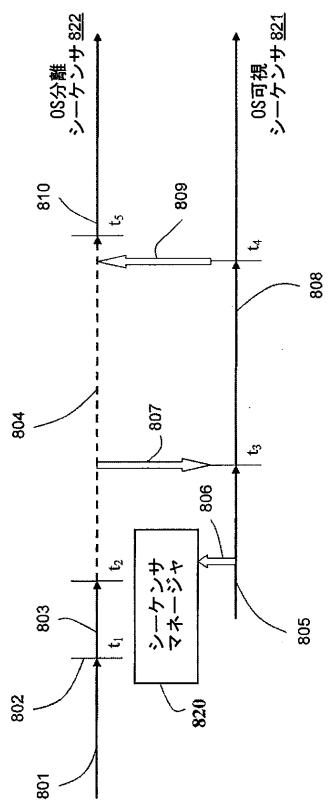
【図7】



【図8】



【図9】



フロントページの続き

- (72)発明者 プラシャント セチ
アメリカ合衆国、95630 カリフォルニア州、フォルサム、シルバー・ホーン ドライブ 23
5
- (72)発明者 シブ カウシク
アメリカ合衆国、97229 オレゴン州、ポートランド、ノースウェスト ハーバー エルエヌ
14116
- (72)発明者 ブライアント ビッグビー
アメリカ合衆国、85259 アリゾナ州、スコットデール、イースト パロミノ ロード 10
886
- (72)発明者 ジョン シエン
アメリカ合衆国、95121 カリフォルニア州、サンノゼ、クエイル ブラフ ブレース 2
303
- (72)発明者 リチャード ハンキンス
アメリカ合衆国、95052 カリフォルニア州、サンタクララ、ミッション カレッジ ブー
ルバード 2200
- (72)発明者 シャン ズー
アメリカ合衆国、97006 オレゴン州、ビーバートン、サウスウェスト ホートン ウェイ
92
- (72)発明者 バイジュ パテル
アメリカ合衆国、97229 オレゴン州、ポートランド、ノースウェスト エルエー カッセル
クレスト レーン 10554
- (72)発明者 ジェーソン プラント
アメリカ合衆国、78703 テキサス州、オースチン、サミット ビュー ブレース 1713
ビー
- (72)発明者 アニル アガールウォル
アメリカ合衆国、97229 オレゴン州、ポートランド、ノースウェスト ケイトリン テラス
1587
- (72)発明者 ジョン レイド
アメリカ合衆国、97229 オレゴン州、ポートランド、ノースウェスト ウォーターフォード
ウェイ 16893

審査官 漆原 孝治

- (56)参考文献 特開平05-265876 (JP, A)
特開平04-097445 (JP, A)
米国特許第05485626 (US, A)
佐藤未来子、外6名、マルチスレッドアーキテクチャ向けOS「Future」におけるプロセス管理、コンピュータシステム・シンポジウム論文集、日本、社団法人情報処理学会、2003年12月11日、Vol. 2003, No. 20, pp. 61 - 70
HANKINS, R. A. et al., Multiple Instruction Stream Processor, Proceedings of the 33rd annual international symposium on Computer Architecture, IEEE, 2006年 6月21日

(58)調査した分野(Int.Cl., DB名)

G06F 9/46 - 9/54