

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2005-165902

(P2005-165902A)

(43) 公開日 平成17年6月23日(2005.6.23)

(51) Int. Cl.⁷

G06F 9/46

F I

G06F 9/46 340B

テーマコード (参考)

5B098

審査請求 未請求 請求項の数 6 O L (全 13 頁)

(21) 出願番号 特願2003-406794 (P2003-406794)

(22) 出願日 平成15年12月5日 (2003.12.5)

(71) 出願人 000003207

トヨタ自動車株式会社

愛知県豊田市トヨタ町1番地

(74) 代理人 100100549

弁理士 川口 嘉之

(74) 代理人 100090516

弁理士 松倉 秀実

(74) 代理人 100098268

弁理士 永田 豊

(74) 代理人 100085006

弁理士 世良 和信

(74) 代理人 100089244

弁理士 遠山 勉

最終頁に続く

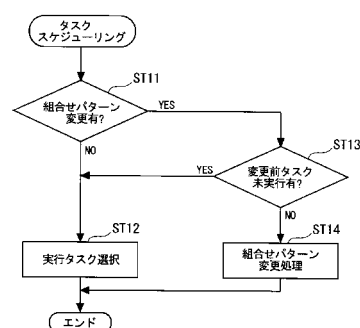
(54) 【発明の名称】 タスク管理システム

(57) 【要約】

【課題】 タスク組合せの切替を迅速に行う。

【解決手段】 タスク組合せを構成するタスクから選択されるタスクを実行するタスク管理システムであって、タスク組合せの切替要求の有無を判定する判定部と、前記切替要求有りと判定された場合、その切替前のタスク組合せを構成するタスクのうち特定タスクの実行完了後に、タスク組合せを切り替える切替部と、前記切替部により切り替えられたタスク組合せを構成するタスクから選択されるタスクを実行する実行部と、を備える。

【選択図】 図4



【特許請求の範囲】**【請求項 1】**

タスク組合せを構成するタスクから選択されるタスクを実行するタスク管理システムであって、

タスク組合せの切替要求の有無を判定する判定部と、
前記切替要求有りと判定された場合、その切替前のタスク組合せを構成するタスクのうち特定タスクの実行完了後に、タスク組合せを切り替える切替部と、

前記切替部により切り替えられたタスク組合せを構成するタスクから選択されるタスクを実行する実行部と、
を備えるタスク管理システム。

10

【請求項 2】

タスク組合せを構成するタスクとそのタスクが特定タスクであるか否かを識別するための識別子との対応関係を格納する記憶部をさらに備え、

前記切替部は、前記切替要求有りと判定された場合、その切替前のタスク組合せを構成するタスクのうち前記識別子に基づいて識別される特定タスクの実行完了後に、タスク組合せを切り替える、請求項 1 に記載のタスク管理システム。

【請求項 3】

前記特定タスクは、タスク組合せの切替時に処理の完了が要求されるタスクである、
請求項 1 又は 2 に記載のタスク管理システム。

【請求項 4】

タスク組合せを構成するタスクから選択されるタスクを実行するタスク管理方法であって、

20

タスク組合せの切替要求の有無を判定する判定ステップ、
前記切替要求有りと判定された場合、その切替前のタスク組合せを構成するタスクのうち特定タスクの実行完了後に、タスク組合せを切り替える切替ステップ、

前記切替部により切り替えられたタスク組合せを構成するタスクから選択されるタスクを実行する実行ステップ、
を備えるタスク管理方法。

【請求項 5】

前記切替ステップは、前記切替要求有りと判定された場合、その切替前のタスク組合せを構成するタスクのうち、タスク組合せを構成するタスクとそのタスクが特定タスクであるか否かを識別するための識別子との対応関係を格納する記憶部中の識別子に基づいて識別される特定タスクの実行完了後に、タスク組合せを切り替える、請求項 4 に記載のタスク管理方法。

30

【請求項 6】

前記特定タスクは、タスク組合せの切替時に処理の完了が要求されるタスクである、
請求項 4 又は 5 に記載のタスク管理方法。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、タスク組合せを構成するタスクから選択されるタスクを実行するタスク管理システムにおいてタスク組合せの切替を迅速に行うための技術に関する。

40

【背景技術】**【0002】**

本出願人は、先に、タスク組合せを構成するタスク（一つ又は複数）から選択されるタスクを実行する発明について出願している（特許文献 1 等）。

【特許文献 1】特願 2003 - 122139

【特許文献 2】特開平 7 - 182183 号公報

【発明の開示】**【発明が解決しようとする課題】**

50

【 0 0 0 3 】

先の出願においては、タスク組合せの切替は、タスク組合せを構成するタスクのうち実行可能なタスクの完了を待ってから行われる。しかしながら、タスクの完了を待ってから切り替えるのでは、迅速に切り替えることができない。また、切替前のタスクを直ちに中断したい場合のように、切替要因によってはタスクを実行してはいけない場合も想定される。

【 0 0 0 4 】

本発明の課題は、タスク組合せの切替を迅速に行うことのできる技術を提供することにある。また、処理すべきでない処理を行わずタスク組合せを切替ることのできる技術を提供することにある。

【課題を解決するための手段】

【 0 0 0 5 】

本発明は、上記課題を解決するためになされたものであり、タスク組合せを構成するタスクから選択されるタスクを実行するタスク管理システムであって、タスク組合せの切替要求の有無を判定する判定部と、前記切替要求有りとは判定された場合、その切替前のタスク組合せを構成するタスクのうち特定タスクの実行完了後に、タスク組合せを切り替える切替部と、前記切替部により切り替えられたタスク組合せを構成するタスクから選択されるタスクを実行する実行部と、を備える構成とした。

【 0 0 0 6 】

本発明によれば、タスク組合せの切替要求有りとは判定された場合、その切替前のタスク組合せ（例えば現在のタスク組合せ）を構成するタスクのうち特定タスクの実行完了後に、タスク組合せを切り替える。すなわち、タスク組合せを構成する全てのタスクの実行完了を待ってからタスク組合せを切り替えるのではなくて、タスク組合せを構成するタスクのうち特定タスクの実行完了を待ってからタスク組合せを切り替える。従って、タスク組合せの切替を迅速に行うことが可能となる。

【 0 0 0 7 】

上記タスク管理システムにおいて、例えば、タスク組合せを構成するタスクとそのタスクが特定タスクであるか否かを識別するための識別子（例えば本実施形態のフラグ）との対応関係を格納する記憶部をさらに備え、前記切替部は、前記切替要求有りとは判定された場合、その切替前のタスク組合せを構成するタスクのうち前記識別子に基づいて識別される特定タスクの実行完了後に、タスク組合せを切り替える。

【 0 0 0 8 】

これは、特定タスクを識別するための手法の一つの例示である。従って、本発明の特定タスクの識別手法はこれに限定されない。他の手法によって特定タスクを識別するようにしてもよい。

【 0 0 0 9 】

このようにすれば、識別子をチェックするだけで特定タスクを識別することができることから、そのような識別子に基づいて識別される特定タスクの実行完了を迅速に把握することが可能となる。また、切替要因に応じて、実行してはいけないタスクに対応する識別子を設定しておけば、切替前のタスクを直ちに中断したい場合のように、切替要因によってはタスクを実行してはいけないタスクが存在するとしても、識別子をチェックするだけでそのようなタスクを識別することができることから、そのようなタスクの実行も防止できる。

【 0 0 1 0 】

上記タスク管理システムにおいて、例えば、前記特定タスクは、タスク組合せの切替時に処理の完了が要求されるタスクである。

これは、特定タスクの一つの例示である。従って、本発明の特定タスクはこれに限定されない。他のタスクを特定タスクとすることも可能である。

【 0 0 1 1 】

また、本発明は方法の発明として次のように特定することもできる。

10

20

30

40

50

タスク組合せを構成するタスクから選択されるタスクを実行するタスク管理方法であって、タスク組合せの切替要求の有無を判定する判定ステップ、前記切替要求有りと判定された場合、その切替前のタスク組合せを構成するタスクのうち特定タスクの実行完了後に、タスク組合せを切り替える切替ステップ、前記切替部により切り替えられたタスク組合せを構成するタスクから選択されるタスクを実行する実行ステップ、を備えるタスク管理方法。

【 0 0 1 2 】

また、本発明はプログラムの発明として次のように特定することができる。

コンピュータに、タスク組合せの切替要求の有無を判定する判定ステップ、前記切替要求有りと判定された場合、その切替前のタスク組合せを構成するタスクのうち特定タスクの実行完了後に、タスク組合せを切り替える切替ステップ、前記切替部により切り替えられたタスク組合せを構成するタスクから選択されるタスクを実行する実行ステップ、を実行させるためのプログラム。

10

【 0 0 1 3 】

また、本発明は、上記プログラムを記録したコンピュータ読み取り可能な記録媒体としても特定できる。

【発明の効果】

【 0 0 1 4 】

本発明によれば、処理を完了しなくてもよいタスクへのリソース割当が省略されることから、タスク組合せの切替を迅速に行うことが可能となる。また、他の効果として、処理すべきでない処理を行わずタスク組合せの切替が可能となる。

20

【発明を実施するための最良の形態】

【 0 0 1 5 】

以下、この発明の一実施形態について図面を参照しながら説明する。なお、以下の一実施形態の全図においては、同一または対応する部分には同一の符号を付す。図 1 に、この一実施形態によるタスク管理システムの構成を示す。

【 0 0 1 6 】

図 1 に示すように、この一実施形態によるタスク管理システムは、PDA(Personal Digital Assistants)やパーソナルコンピュータなどの情報処理端末 100 により実現される。

30

【 0 0 1 7 】

この情報処理端末 100 は、中央処理装置(CPU)110、例えばハードディスク装置などの補助記憶装置 120、例えば書き換え可能メモリ(RAM)などのメモリ 130、例えば液晶ディスプレイ(LCD)などの画像表示装置やスピーカなどの音声出力装置(いずれも図示せず)などを有して構成されている。これらのそれぞれの要素は、例えばバスを介して互いに接続されている。

【 0 0 1 8 】

また、補助記憶装置 120 には、リアルタイムOS(Operating System)と、このリアルタイムOSをベースとして実行可能な種々のアプリケーション、すなわち優先度が設定された複数のタスクなどと、後述するタスク情報テーブルなどがインストールされている。

40

【 0 0 1 9 】

これらのリアルタイムOSなどのソフトウェアは、補助記憶装置 120 からメモリ 130 に、必要に応じて、適宜読み出される。図 1 は、これらのリアルタイムOSなどのソフトウェアがメモリ 130 に読み出された状態を示す。

【 0 0 2 0 】

また、リアルタイムOSは、複数のタスクのうちの最も高い優先度が設定されているタスクを実行する優先度制御機能(または、タスク管理機能)を有するオペレーティングシステムである。優先度制御機能は、リアルタイムOSが有するスケジューラ 131 によって実現される。

50

【 0 0 2 1 】

スケジューラ 1 3 1 は、それぞれのタスクに設定されている優先度に基づいて複数のタスクのうちの最も高い優先度が設定されているタスクを実行するためのものである。また、スケジューラ 1 3 1 は、優先度が設定されている複数のタスクとして把握されるアプリケーション群や、各種デバイスの制御などの処理における、例えばタイミング処理を実行するか否かを決定可能に構成されている。

【 0 0 2 2 】

具体的には、スケジューラ 1 3 1 は、終了または待ち状態でないタスクのうちの最も高い優先度が設定されているタスクを継続して実行（処理）する。このとき、リアルタイム OS においては、優先度が低く設定されているタスクは、全く動作しない可能性もある。また、スケジューラ 1 3 1 は、それぞれのタスクの状態、具体的には実行可能状態、実行状態、待ち状態などを管理するためのものである。

【 0 0 2 3 】

スケジューラ 1 3 1 によって優先度制御がされる複数のタスクとしては、スケジューラタスク 1 3 2、時間起動通知用タスク 1 3 3、および、Q o S (Quality of Service) タスク 1 3 4 などがある。これらのタスクには、優先度が設定されている。なお、スケジューラタスク 1 3 2 には、最も高い優先度（他のタスク 1 3 3、1 3 4 などと比較して高い優先度）が設定される。

【 0 0 2 4 】

また、スケジューラタスク 1 3 2 は、他のタスク（Q o S タスク 1 3 4）の実行制御を行うためのタスクである。スケジューラタスク 1 3 2 は、他のタスクの実行制御として、Q o S タスク 1 3 4 の中から実行タスクの決定、実行タスクの優先度の変更、および、実行タスクの所要リソース量の制御（リソース割り当て）などを行う。

【 0 0 2 5 】

また、時間起動通知用タスク 1 3 3 は、スケジューラタスク 1 3 2 を実行状態（Running 状態）へ遷移させるためのものである。そのため、時間起動通知用タスク 1 3 3 は、所定期間（一定周期または設定時間）ごとに、イベントフラグまたはセマフォなどの所定通知をスケジューラタスク 1 3 2 へ通知する。所定通知を受けると、スケジューラタスク 1 3 2 は、実行状態へ遷移して他のタスク（Q o S タスク 1 3 4）の実行制御を行う。なお、スケジューラタスク 1 3 2 において、実行状態にない場合には、リアルタイム OS におけるスケジューラ 1 3 1 により、Q o S タスク 1 3 4 が制御される。

【 0 0 2 6 】

また、Q o S タスク 1 3 4 は、その所要リソース量を制御可能な可変タスクである。そして、所要リソース量が最低限保証すべき保証部分と制御可能部分とに分けられているため、最低限の品質（保証部分）を保持可能に構成されている。

【 0 0 2 7 】

また、タスク組合せ管理テーブル 1 3 6 は、アプリケーションの構成情報や、それぞれのタスクの所要リソース量情報などに基づいて、C P U 1 1 0 により抽出されたタスクの組合せを格納するためのものである。このタスク組合せ管理テーブル 1 3 6 の一例を図 2 に示す。

【 0 0 2 8 】

すなわち、タスク組合せ管理テーブル 1 3 6 におけるタスク組合せのデータは、図 2 A に示すように、抽出された Q o S タスク 1 3 4 の組合せ（タスク組合せ）を示し、タスク組合せを識別するためのパターン番号と、そのパターン番号に属するタスクを組み合わせたりスト（タスクリスト）とから構成される。

【 0 0 2 9 】

また、タスクリストには、それぞれの Q o S タスク 1 3 4 に固有のタスク識別子 I D （タスク I D）とフラグ flag が格納されており、必要に応じて、タスク情報テーブル 1 3 5 等から Q o S タスク 1 3 4 の構成情報やフラグ flag が読み込まれる。フラグ flag は、タスク組合せの切替時に処理の完了を要求されているか否かを示すフラグである。このタスク

10

20

30

40

50

組合せ管理テーブル 136 中のタスクリストにおけるタスク ID とフラグ flag との対応関係が、本発明のタスク組合せを構成するタスクとそのタスクが特定タスクであるか否かを識別するための識別子との対応関係に相当する。QoS タスク 134 の詳細情報は、タスク情報テーブル 135 に格納され、タスク ID 135a を識別子として抽出可能となっている。

【0030】

そして、図 2B に示すように、この一実施形態によるタスク組合せ管理テーブル 136 は、図 2A に示すパターン番号とこの番号に属するタスクリストとの組の少なくとも 1 つから構成されており、パターン番号により、相互にタスクリストを区別可能に格納されている。

10

【0031】

そして、複数のタスクリストがパターン番号により区別されて格納されたタスク組合せ管理テーブル 136 から、アプリケーションにより所定のタスクリストが選択され、選択されたパターン番号に属するタスクリストのそれぞれのタスク ID 及びフラグ flag に基づいて、タスク情報テーブル 135 から所定の QoS タスク 134 が読み出され、アプリケーションにより処理が実行される。

【0032】

また、この一実施形態によるタスク組合せ管理テーブル 136 は、アプリケーションごとに生成され、たとえば補助記憶装置 120 に格納される。また、一旦生成されたタスク組合せ管理テーブル 136 は、アプリケーションごとに関連づけられて、アプリケーションの起動などの必要に応じて、補助記憶装置 120 からメモリ 130 に読み出される。そして、アプリケーションにおけるタスク組合せは、この読み出されたタスク組合せ管理テーブル 136 から抽出される。

20

【0033】

このように、あらかじめアプリケーションごとに、タスク組合せ管理テーブル 136 を生成しておき、タスクリストから必要なタスク組合せを選択して、実行することにより、アプリケーションの実行に基づくタスクの実行における実行の可否の計算処理を、タスクの登録ごとに行う必要がなくなるので、時間のロスを無くして、タスクの起動や切替をスムーズに行うことができる。

【0034】

なお、タスク組合せ管理テーブル 136 を、補助記憶装置 120 に格納しておき、後述するパターン番号が選択されたときに、必要なパターンのみを読み込むようにすることも可能である。

30

【0035】

また、タスク情報テーブル 135 は、QoS タスク 134 に関する情報を管理するためのテーブルである。タスク情報テーブル 135 は、タスク識別子（タスク ID）135a と、このタスク ID 135a に属する各種情報項目、例えば起動タイミング 135b や実行割り当て時間 135c などの項目とにより構成されている。なお、タスク ID 135a は、QoS タスクを識別するための識別子 ID であり、起動タイミング 135b は、QoS タスクにおける実行開始時点から、次の実行開始時点までの時間（周期）である。

40

【0036】

そして、ある QoS タスクが実行されると、設定された周期の間、この QoS タスクが新たに実行されることはない。また、実行割り当て時間 135c は、QoS タスクに割り当てられたリソース量であり、この一実施形態においては、例えば CPU の使用時間である。

【0037】

以上のようにして、この一実施形態によるタスク管理システムが構成されている。なお、上述したタスク管理システムは、あくまでも一例であり、必ずしも上述した構成に限られるものではなく、そのほかのタスク管理システム、たとえばスケジューラタスクが設定されておらず、全てのタスクスケジュールにおいて、リアルタイム OS におけるスケジュー

50

ーラを利用するようにしたタスク管理システムを採用することも可能である。

【0038】

次に、以上のように構成されたタスク管理システムの動作について図面を参照しつつ説明する。図3に、この一実施形態による新規タスクの登録要求動作を示す。

(新規タスク登録要求)

まず、アプリケーションの起動時においては、このアプリケーションの構成情報に基づいて、あらかじめ抽出された、必要となるQoS制御のタスク組合せの全てのパターンに対して、タスク予約判定を行った後に登録を行う、いわゆる新規タスクの登録要求処理が実行される。以下に、この新規タスクの登録要求処理について説明する。

【0039】

すなわち、図3に示すように、この一実施形態による新規タスク登録要求処理においては、まず、ステップST1において、タスク組合せのパターンを抽出する処理を実行する。具体的には、アプリケーションの構成情報と、それぞれのQoSタスク134の所要リソース量情報などに基づいて、タスク組合せ管理テーブル136から、組み合わせられて実行されることにより一機能となり得るタスク組合せが抽出される。その後、ステップST2に移行する。

【0040】

次に、ステップST2においては、タスク組合せのパターンが抽出されたか否か、すなわち抽出パターンが存在するか否かの判定が行われる。その結果、タスク組合せの抽出パターンが存在しなかった場合、ステップST3に移行して、アプリケーションにより、新

10

20

【0041】

他方、ステップST2において、タスク組合せのパターンが抽出された場合、すなわち、アプリケーションに基づくタスク組合せに合致する抽出パターンが存在した場合、ステップST4に移行する。

【0042】

ステップST4においては、抽出されたパターンが、新規タスクとして登録可能であるか否かの判断が行われる。具体的には、例えば、抽出されたタスクの組合せについて、例えば、動作環境における負荷とQoSの検証とが行われ、負荷が所定の基準値を下回りかつQoSが所定の基準値を上回っている場合に、実行可能と判定され、そうでない場合に

30

【0043】

そして、ステップST4において、タスク組合せが実行不可能である場合には、抽出パターンが登録不可とされ、ステップST5に移行して、登録不可の通知が行われる。この登録不可の通知は、サービス品質(QoS)を設定可能なタスク管理システムにおいて、タスクの受付を行う部分に対して行われる。以上により一連の新規タスク登録処理が終了する。

【0044】

他方、ステップST4における判断の結果、タスク組合せを実行可能である場合に、抽出パターンが登録可能であるとされ、ステップST1に移行する。そして、再度、アプリケーションの構成情報に基づいて、タスク組合せのパターンの抽出が行われ、ステップST2において、上述と同様にして、抽出パターンの存否が判断される。

40

【0045】

これらのステップST1、ステップST2およびステップST4のプロセスは、登録不可の抽出パターンが表出するか、抽出パターンが存在しなくなるまで、繰り返し実行される。

【0046】

その後、ステップST2においてタスク組合せのパターンが全て抽出された段階で、ステップST3に移行し、このステップST3において、アプリケーションにより新規タスクが登録され、新規タスクの登録処理が終了する。

50

【 0 0 4 7 】

また、繰り返し登録処理が実行されている間に、実行ができないタスク組合せが現れて、ステップ S T 4 において抽出パターンの登録ができない場合、ステップ S T 5 においてスケジューラタスク 1 3 2 に登録不可の通知がされて、新規タスクの登録処理が終了する。

【 0 0 4 8 】

(タスクスケジューリング)

次に、以上のようにして、プログラムの起動時に新規タスクの登録処理が行われた後における、アプリケーションの実行時のタスク管理について説明する。すなわち、アプリケーションの実行中においては、動作させるタスクが新たに実行されたり、タスク間において切替が行われたりする、いわゆるタスクスケジューリングが行われる。以下に、この一実施形態によるタスクスケジューリングについて説明する。図 4 に、この一実施形態によるタスクスケジューリング処理を示す。

【 0 0 4 9 】

すなわち、図 4 に示すように、この一実施形態によるタスクスケジューリングにおいては、まず、ステップ S T 1 1 において、組合せパターン変更の有無についての判定、すなわち、タスク組合せパターンの切替要求の有無についての判定が行われる(本発明の判定部に相当)。この判定は、例えば、現在のタスク組合せパターン番号と切替要求のあったタスク組合せパターン番号とを比較することで行う。そして、両番号が異なっていれば、組合せパターン変更有り、すなわち、タスク組合せパターンの切替要求有と判定される(ステップ S T 1 1 : Yes)。

【 0 0 5 0 】

このタスク組合せのパターン変更無しと判定された場合(ステップ S T 1 1 : No)、ステップ S T 1 2 に移行して、現在のタスク組合せを構成するタスクから実行タスクが選択され、この選択されたタスクが C P U 1 1 0 によりアプリケーションに基づいて実行される。

【 0 0 5 1 】

他方、タスク組合せパターンの変更有りと判定された場合(ステップ S T 1 1 : Yes)、ステップ S T 1 3 に移行して、その変更前(切替前)のタスク組合せ(例えば現在のタスク組合せ)を構成するタスク(一つのこともあるし、複数のこともある)の中に、タスク組合せの切替時に処理の完了を要求されているタスク(本発明の特定タスクに相当)があるか否かについての判定が行われる。この判定のために、タスク組合せ管理テーブル 1 3 6 中のフラグ flag が参照される。上述の通り、フラグ flag は、タスクごとに登録されており、しかもタスク組合せの切替時に処理の完了を要求されているタスク(本発明の特定タスクに相当)であるか否かを示すフラグである。従って、タスク組合せ管理テーブル 1 3 6 中のフラグ flag が参照されることで、その変更前(切替前)のタスク組合せ(例えば現在のタスク組合せ)を構成するタスクの中に、タスク組合せの切替時に処理の完了を要求されているものがあるか否かについての判定を行うことが可能となっている。

【 0 0 5 2 】

具体的には、次のようにして判定される。まず、その変更前(切替前)のタスク組合せ(例えば現在のタスク組合せ)を構成するタスクの中に、実行されていないタスク(または、中断されているタスク)が存在するか否かの判定が行われる。そして、実行されていないタスク(または、中断されているタスク)が存在すると判定された場合、その実行されていないタスク(または、中断されているタスク)に対応するフラグ flag が参照される。これにより、判定対象タスクを、実行されていないタスク(または、中断されているタスク)に限っている。従って、すでに実行完了したタスクについての判定を行うことが無くなることから、判定速度を高めることが可能となる。

【 0 0 5 3 】

このステップ S T 1 3 において、変更前(切替前)のタスク組合せ(例えば現在のタスク組合せ)を構成するタスクの中に、タスク組合せの切替時に処理の完了を要求されてい

10

20

30

40

50

るタスク（本発明の特定タスクに相当）が存在する場合（ステップ S T 1 3 : Yes）、ステップ S T 1 2 に移行して、実行しているタスク組合せが C P U 1 1 0 により継続して実行される。すなわち、現在のタスク組合せを構成するタスクから実行タスクが選択され、この選択されたタスクが C P U 1 1 0 によりアプリケーションに基づいて継続実行される。S T 1 1 から S T 1 4 の処理はタスクスケジューリングによるタイミングが到来するごとに繰り返される。これにより、やがて、変更前（切替前）のタスク組合せ（例えば現在のタスク組合せ）を構成するタスクの中に、タスク組合せの切替時に処理の完了を要求されているタスク（本発明の特定タスクに相当）が存在しなくなる（ステップ S T 1 3 : No）。そして、そのようなタスクが存在しなくなれば（ステップ S T 1 3 : No）、ステップ S T 1 4 に移行して、組み合わせパターンを変更（切替）する組み合わせパターン変更処理が実行される。すなわち、実行が中止等されている全てのタスクについて実行完了を待たない状況となった場合に、タスク組合せが切り替えられる。そして、S T 1 1 から S T 1 4 の処理はタスクスケジューリングによるタイミングが到来するごとに繰り返されることから、その切り替え後のタスク組合せを構成するタスクから実行タスクが選択され（S T 1 2）、この選択されたタスクが C P U 1 1 0 によりアプリケーションに基づいて継続実行される（本発明の実行部に相当）。 10

【 0 0 5 4 】

以上のようにして、この一実施形態による、アプリケーションの実行に伴うタスクにおけるスケジューリングが行われる。 20

（タスク組合せ登録要求処理）

次に、以上のようにして、新規タスク登録や、タスクスケジューリングが行われる際に用いられるタスク組合せ管理テーブル 1 3 6 への、タスクの登録方法について、説明する。図 5 に、この一実施形態によるタスク組合せの登録処理を示す。

【 0 0 5 5 】

図 5 に示すように、アプリケーションなどから、タスク組合せの登録要求がされた場合には、まず、ステップ S T 2 1 において、登録要求された複数のタスクが登録可能であるか否かの判断が行われる。

【 0 0 5 6 】

ステップ S T 2 1 において、登録要求された複数のタスクが、組合せとして Q o S を維持しつつ実行することができない場合、すなわち Q o S 制御が実行不可の場合（ステップ S T 2 1 : No）、タスク組合せ管理テーブル 1 3 6 への登録処理は終了する。 30

【 0 0 5 7 】

他方、ステップ S T 2 1 において、タスク組合せが Q o S を維持しつつ実行可能な場合、すなわち登録要求された複数のタスクが登録可能である場合（ステップ S T 2 1 : Yes）、ステップ S T 2 2 に移行して、タスク組合せ管理テーブル 1 3 6 に登録可能であるか否かの判断が行われる。

【 0 0 5 8 】

そして、ステップ S T 2 2 における登録の可否に関して、登録ができないと判断された場合、ステップ S T 2 3 に移行して、登録不可通知が行われる。なお、この登録不可通知は、サービス品質を設定可能なタスク管理における管理メカニズムにおけるタスクの受付を行う部分（例えば、スケジューラやスケジューラタスク）に対して行われる。 40

【 0 0 5 9 】

他方、ステップ S T 2 2 において、タスク組合せ管理テーブル 1 3 6 に登録可能と判断された場合には、ステップ S T 2 4 に移行して、タスク組合せ管理テーブル 1 3 6 に、所定の識別可能なパターン番号が生成され、このパターン番号と、これに属する上述の少なくとも 1 つのタスクにおけるタスク I D 及びフラグ flag からなるタスクリストとが登録される。そして、このようにして登録された各種タスク組合せが、上述したようにアプリケーションにより適切に選択され、実行される。

【 0 0 6 0 】

その後、ステップ S T 2 5 に移行して、この登録されたパターン番号が、Q o S を設定 50

可能なタスク管理における管理メカニズムにおけるタスクを選択可能に構成されたスケジューラやスケジューラタスクに通知され、この一実施形態によるタスク組合せ登録の処理が終了する。以上のようにして、アプリケーションの起動時に、実行可能な範囲で、全てのQoSタスクが、タスク組合せ管理テーブル136に登録される。

【0061】

以上説明したように、この一実施形態によれば、アプリケーションの起動時に、このアプリケーションに用いられる全てのQoSタスクのうちの、登録可能なQoSタスク組合せを、タスク組合せ管理テーブル136に登録し、アプリケーションの実行時におけるタスクの実行時には、タスク組合せ管理テーブル136に登録されたタスクリストに基づいて、必要なタスク組合せのパターン番号を実行することにより、実行するアプリケーションのQoS制御のタスク組合せのパターンをあらかじめ抽出しておくことができ、QoSタスク134の処理に関して、そのタスク組合せが実行可能であるか否かを判断する時間が不要になる。

10

【0062】

したがって、この判断に要する時間のロスを防止することができ、タスク組合せの切替時間を大幅に短縮することができる。また、計算時間が長くなることによりタスク制御が不能になったり、アプリケーションによるタスクの登録ができなくなったりすることを防止することができ、この抽出に基づいたタスクの管理、および実行中のタスク組合せをスムーズに変更することが可能となるので、アプリケーションのタスク予約の受付を確実に実行することができるとともに、効率的なスケジューリングを実行することができる。

20

【0063】

また、処理を完了しなくてもよいタスク（タスク組合せの切替時に処理の完了を要求されていないタスク）への割当が省略されることから、タスク組合せの切替を迅速に行うことが可能となる。また、他の効果として、処理すべきでない処理（タスク組合せの切替時に処理の完了を要求されていないタスクの実行等）を行わずタスク組合せの切替が可能となる。

【0064】

以上、この発明の一実施形態について具体的に説明したが、この発明は、上述の一実施形態に限定されるものではなく、この発明の技術的思想に基づく各種の変形が可能である。

30

【0065】

例えば、上述の一実施形態において挙げたシステムの構成はあくまでも例に過ぎず、必要に応じてこれと異なるシステム構成を採用することも可能である。

また、例えば上述の一実施形態においては、リソース量として、使用環境におけるCPU110の使用時間を採用しているが、CPUのほかに記憶装置やメモリ、ビデオメモリ（図示せず）の使用時間および記憶容量、ネットワークの通信速度やネットワークサーバの記憶容量、メモリ容量など、特定のタスクを実行するために必要なハードウェア資源およびソフトウェア資源の全てを選択的に用いることが可能である。

【産業上の利用可能性】

【0066】

本発明によれば、処理を完了しなくてもよいタスクへの割当が省略されることから、タスク組合せの切替を迅速に行うことが可能となる。また、他の効果として、処理すべきでない処理を行わずタスク組合せの切替が可能となる。

40

【図面の簡単な説明】

【0067】

【図1】この発明の一実施形態によるタスク管理システムを示すブロック図である。

【図2】この発明の一実施形態によるタスク管理に用いられるタスク組合せ管理テーブルを示す図である。

【図3】この発明の一実施形態によるタスク管理方法の新規タスク登録方法を説明するためのフローチャートである。

50

【図 4】この発明の一実施形態によるタスク管理方法のタスクの切替方法を説明するためのフローチャートである。

【図 5】この発明の一実施形態によるタスク管理方法のタスク組合せ管理テーブルへの登録方法を示すフローチャートである。

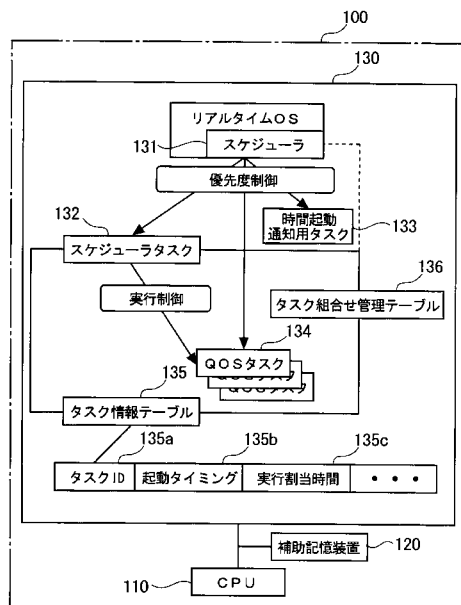
【符号の説明】

【 0 0 6 8 】

- 1 0 0 情報処理端末
- 1 1 0 中央処理装置 (C P U)
- 1 2 0 補助記憶装置
- 1 3 0 メモリ
- 1 3 1 スケジューラ
- 1 3 2 スケジューラタスク
- 1 3 3 時間起動通知用タスク
- 1 3 4 Q o S タスク
- 1 3 5 タスク情報テーブル
- 1 3 5 a タスク I D
- 1 3 5 b 起動タイミング
- 1 3 5 c 起動割当時間
- 1 3 6 タスク組合せ管理テーブル

10

【 図 1 】



【 図 2 】

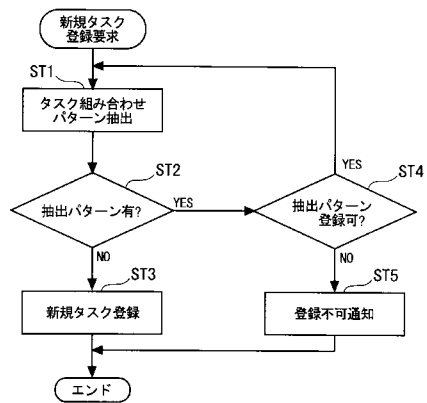
(A)

パターン番号	タスクID1	flag	タスクID2	flag	タスクIDn	flag
					...	

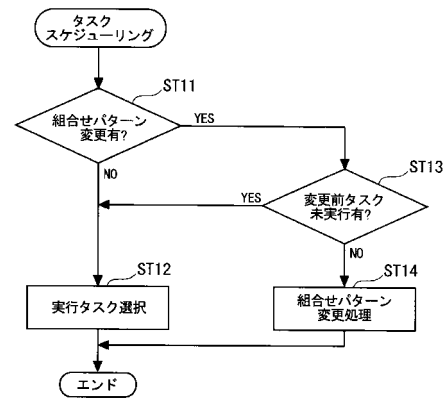
(B)

パターン番号	タスクID	flag	タスクID	flag	タスクID	flag
1	タスクID	flag	タスクID	flag	タスクID	flag
2	タスクID	flag	タスクID	flag	タスクID	flag
...						
n	タスクID	flag	タスクID	flag	タスクID	flag

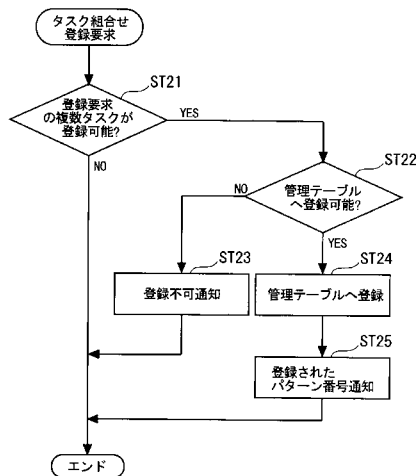
【 図 3 】



【 図 4 】



【 図 5 】



フロントページの続き

(72)発明者 阿部 睦

東京都目黒区中目黒 2 丁目 1 番 2 3 号 株式会社トヨタ I T 開発センター内

F ターム(参考) 5B098 GA04 GC01