

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5379906号
(P5379906)

(45) 発行日 平成25年12月25日 (2013.12.25)

(24) 登録日 平成25年10月4日 (2013.10.4)

(51) Int. Cl. F I
G O 6 F 9/52 (2006.01) G O 6 F 9/46 4 7 5 B

請求項の数 17 (全 40 頁)

(21) 出願番号	特願2012-501735 (P2012-501735)	(73) 特許権者	000005108
(86) (22) 出願日	平成23年2月10日 (2011.2.10)		株式会社日立製作所
(86) 国際出願番号	PCT/JP2011/052880		東京都千代田区丸の内一丁目6番6号
(87) 国際公開番号	W02011/105226	(74) 代理人	110000176
(87) 国際公開日	平成23年9月1日 (2011.9.1)		一色国際特許業務法人
審査請求日	平成24年3月15日 (2012.3.15)	(72) 発明者	新井 利明
(31) 優先権主張番号	特願2010-190256 (P2010-190256)		神奈川県横浜市戸塚区吉田町292番地
(32) 優先日	平成22年8月27日 (2010.8.27)		株式会社日立製作所 システム開発研究所
(33) 優先権主張国	日本国 (JP)		内
(31) 優先権主張番号	特願2010-42848 (P2010-42848)	(72) 発明者	天野 光司
(32) 優先日	平成22年2月26日 (2010.2.26)		神奈川県横浜市戸塚区吉田町292番地
(33) 優先権主張国	日本国 (JP)		株式会社日立製作所 システム開発研究所
			内

最終頁に続く

(54) 【発明の名称】 計算機、計算機制御方法、および計算機制御プログラム

(57) 【特許請求の範囲】

【請求項1】

他の計算機と同じタスクを処理する計算機であって、
 前記他の計算機と同期をとってタスクを処理する同期プロセッサと、
 前記他の計算機と同期をとらずにタスクを処理する非同期プロセッサと、
前記同期プロセッサに割り当てられたタスクの状態を管理する同期タスク管理テーブル
 と、前記非同期プロセッサに割り当てられたタスクの状態を管理する非同期タスク管理テ
ーブルと、

前記同期プロセッサ及び前記非同期プロセッサにタスクを割り当てるディスパッチャを
 有しており、

当該計算機と前記他の計算機は前記同期タスク管理テーブルの内容を、タスク制御用共
有メモリ区画を介して相互に通信して交換しており、

前記ディスパッチャは、

タスクが前記他の計算機と同期をとって処理すべきタスクか否かを判断し、当該判断結
 果に基づいて、前記他の計算機と同期をとりながら処理すべき同期タスクは前記同期プロ
 セッサに割当て、前記他の計算機との同期が不要な非同期タスクは前記非同期プロセッサ
 に割当て、同期タスク又は非同期タスクのいずれとしてでも処理可能な同期・非同期タス
クについて前記同期プロセッサ又は前記非同期プロセッサのどちらに割り当てるかを判定
して、該当プロセッサに割当て、

前記同期プロセッサが実行している実行中同期タスクを、他のタスクに切り替える切替

ポイントを検出した場合に、前記実行中同期タスクの状態を待機中に変更し、

当該計算機の前記同期タスク管理テーブルの内容と前記他の計算機の前記同期タスク管理テーブルの内容とに基づいて、当該計算機の前記同期タスク管理テーブルに実行可能タスクとして登録されている複数の同期タスクの内の一を選択して前記同期プロセッサに実行させる計算機。

【請求項 2】

請求項 1 記載の計算機であって、

前記ディスパッチャは、当該計算機の前記同期タスク管理テーブルと、前記他の計算機の前記同期タスク管理テーブルの双方に、実行可能タスクとして登録されている同期タスクを選択して、前記同期プロセッサに実行させる計算機。

10

【請求項 3】

請求項 2 記載の計算機であって、

前記ディスパッチャは、前記同期プロセッサが実行している実行中同期タスクを非同期プロセッサに割り当てる切替ポイントを検出した場合に、前記実行中同期タスクを前記同期タスク管理テーブルから削除して、前記非同期タスク管理テーブルに当該同期タスクを実行可能タスクとして登録する計算機。

【請求項 4】

請求項 3 記載の計算機であって、

前記ディスパッチャは、前記同期プロセッサから前記非同期プロセッサへ割当を変更した前記同期タスクを再び前記同期プロセッサに割り当てる場合に、当該同期タスクを前記非同期タスク管理テーブルから削除して、前記同期タスク管理テーブルに実行可能タスクとして登録し、

20

前記他の計算機との間で前記同期タスク管理テーブルの内容を交換して、前記計算機の前記同期タスク管理テーブルの内容と前記他の計算機の前記同期タスク管理テーブルの内容とに基づいて、前記計算機の前記同期タスク管理テーブルに実行可能タスクとして登録されている複数の同期タスクの内の一を選択して前記同期プロセッサに実行させる計算機。

【請求項 5】

請求項 4 記載の計算機であって、

前記切替ポイントは、前記実行中同期タスクが呼び出した、オペレーションシステムへのシステムコールである計算機。

30

【請求項 6】

請求項 4 記載の計算機であって、

前記切替ポイントは、前記実行中同期タスクが呼び出した、他のタスクから値を取得、もしくは埋め込みをするための命令である計算機。

【請求項 7】

請求項 4 記載の計算機であって、

前記非同期タスク管理テーブルは、複数の同期タスクを当該非同期タスク管理テーブルに割り当てられる順番に管理しており、

前記ディスパッチャは、前記順番に基づいて前記非同期プロセッサに実行させる同期タスクを選択する計算機。

40

【請求項 8】

請求項 4 記載の計算機であって、

前記他の計算機から前記切替ポイントを検知した旨の通知を所定の時間内に受信しない場合に、当該他の計算機に同期が外れている旨の通知を送信する計算機。

【請求項 9】

請求項 1 記載の計算機であって、

前記同期プロセッサ又は前記非同期プロセッサのいずれにも実行され得るタスクについて前記ディスパッチャは、当該タスクの起動を指示した親タスクが割り当てられているプロセッサの種別に基づいて、当該親タスクが同期プロセッサに割り当てられている場合に

50

は当該タスクを同期プロセッサに、当該親タスクが非同期プロセッサに割り当てられている場合には非同期プロセッサに当該タスクを割り当てる計算機。

【請求項 10】

請求項 9 記載の計算機であって、
更に記憶装置を有しており、

前記記憶装置は、前記同期プロセッサ又は前記非同期プロセッサのいずれにも実行され得るタスクについて、当該同期プロセッサおよび当該非同期プロセッサ各々について、当該タスクの実行契機を判断するために用いられる閾値を格納しており、

前記同期プロセッサ用の前記閾値は、前記非同期プロセッサ用の前記閾値とは異なる値に設定されている計算機。

10

【請求項 11】

請求項 10 記載の計算機であって、

前記同期プロセッサ又は前記非同期プロセッサのいずれにも実行され得るタスクとは、当該計算機が有するメモリの回収タスクである計算機。

【請求項 12】

請求項 11 記載の計算機であって、

前記メモリの回収タスクの実行契機を判断するために用いられる閾値は、メモリの空き容量についての閾値であり、前記同期プロセッサ用の閾値の方が前記非同期プロセッサ用の閾値より小さく設定されている計算機。

【請求項 13】

他の計算機と同じタスクを処理する計算機であって、

前記他の計算機と同期をとってタスクを処理する同期プロセッサと、

前記他の計算機と同期をとらずにタスクを処理する非同期プロセッサと、

当該同期プロセッサ及び当該非同期プロセッサにタスクを割り当てるディスパッチャを有しており、

前記ディスパッチャによって前記同期プロセッサ又は前記非同期プロセッサのいずれにも実行され得るタスクには、当該同期プロセッサおよび当該非同期プロセッサ各々について、当該タスクの実行契機を判断するために用いられる閾値が設定されており、

前記同期プロセッサ用の前記閾値は、前記非同期プロセッサ用の前記閾値とは異なる値に設定されており、

20

30

前記同期プロセッサは前記同期プロセッサ用の前記閾値に基づいて、前記タスクの実行契機を判断し、

前記非同期プロセッサは前記非同期プロセッサ用の前記閾値に基づいて、前記タスクの実行契機を判断する計算機。

【請求項 14】

請求項 13 記載の計算機であって、

前記同期プロセッサ又は前記非同期プロセッサのいずれにも実行され得るタスクとは、当該計算機が有するメモリの回収タスクである計算機。

【請求項 15】

請求項 14 記載の計算機であって、

前記メモリの回収タスクの実行契機を判断するために用いられる閾値は、メモリの空き容量についての閾値であり、前記同期プロセッサ用の閾値の方が前記非同期プロセッサ用の閾値より小さく設定されている計算機。

40

【請求項 16】

他の計算機と同じタスクを処理する計算機の制御方法であって、

前記他の計算機と同期をとってタスクを処理する同期プロセッサと、前記他の計算機と同期をとらずにタスクを処理する非同期プロセッサと、前記同期プロセッサに割り当てられたタスクの状態を管理する同期タスク管理テーブルと、前記非同期プロセッサに割り当てられたタスクの状態を管理する非同期タスク管理テーブルと、前記同期プロセッサ及び前記非同期プロセッサにタスクを割り当てるディスパッチャとを有し、前記他の計算機と

50

の間で、前記同期タスク管理テーブルの内容を、タスク制御用共有メモリ区画を介して相互に通信して交換している計算機において、前記ディスパッチャが、

タスクが前記他の計算機と同期をとって処理すべきタスクか否かを判断し、当該判断結果に基づいて、前記他の計算機と同期をとりながら処理すべき同期タスクは前記同期プロセッサに割当て、前記他の計算機との同期が不要な非同期タスクは前記非同期プロセッサに割当て、同期タスク又は非同期タスクのいずれとしてでも処理可能な同期・非同期タスクについて前記同期プロセッサ又は前記非同期プロセッサのどちらに割り当てるかを判定して、該当プロセッサに割当て、

前記同期プロセッサが実行している実行中同期タスクを、他のタスクに切り替える切替ポイントを検出した場合に、前記実行中同期タスクの状態を待機中に変更し、

当該計算機の前記同期タスク管理テーブルの内容と前記他の計算機の前記同期タスク管理テーブルの内容とに基づいて、当該計算機の前記同期タスク管理テーブルに実行可能タスクとして登録されている複数の同期タスクの内の一を選択して前記同期プロセッサに実行させる計算機制御方法。

【請求項 17】

他の計算機と同じタスクを処理する計算機であって、前記他の計算機と同期をとってタスクを処理する同期プロセッサと、前記他の計算機と同期をとらずにタスクを処理する非同期プロセッサと、前記同期プロセッサに割り当てられたタスクの状態を管理する同期タスク管理テーブルと、前記非同期プロセッサに割り当てられたタスクの状態を管理する非同期タスク管理テーブルと、前記同期プロセッサ及び前記非同期プロセッサにタスクを割り当てるディスパッチャとを有し、前記他の計算機との間で、前記同期タスク管理テーブルの内容を、タスク制御用共有メモリ区画を介して相互に通信して交換している計算機において、前記ディスパッチャに、

タスクが前記他の計算機と同期をとって処理すべきタスクか否かを判断し、当該判断結果に基づいて、前記他の計算機と同期をとりながら処理すべき同期タスクは前記同期プロセッサに割当て、前記他の計算機との同期が不要な非同期タスクは前記非同期プロセッサに割当て、同期タスク又は非同期タスクのいずれとしてでも処理可能な同期・非同期タスクについて前記同期プロセッサ又は前記非同期プロセッサのどちらに割り当てるかを判定して、該当プロセッサに割当てる処理と、

前記同期プロセッサが実行している実行中同期タスクを、他のタスクに切り替える切替ポイントを検出した場合に、前記実行中同期タスクの状態を待機中に変更する処理と、

当該計算機の前記同期タスク管理テーブルの内容と前記他の計算機の前記同期タスク管理テーブルの内容とに基づいて、当該計算機の前記同期タスク管理テーブルに実行可能タスクとして登録されている複数の同期タスクの内の一を選択して前記同期プロセッサに実行させる処理と

を実行させる計算機制御プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、計算機、計算機制御方法、および計算機制御プログラムに関するものである。

【背景技術】

【0002】

計算機システムにおけるシステム稼働時間を長時間化するために、計算機を多重化する技術が存在している。この技術は、例えばデュアルシステムなどにより、ハードウェアの障害に際してもシステム内の処理を継続することが出来るものである。

【0003】

また、システム停止時間を最小化するための計算機として、フォールト・トレラントコンピュータ (Fault tolerant computer) がある。このフォールト・トレラントコンピュータの技術は、ハードウェアを多重化し、同一プログラムを複数

10

20

30

40

50

のCPUで動作させて、CPUの障害時にも処理を継続可能とするものである。例えば、本出願人は分散配置に適したフォールトトレラントシステムの構成を提案している（特許文献1参照）。

【0004】

こうした従来のデュアルシステムやフォールトトレラントシステムにおいては、その実現に際して、クロック同期（ロックステップ方式）などを必要とする。そのため、専用ハードウェアに各計算機の通信インターフェースが直接接続されている計算機構成が一般的となっていた。

【先行技術文献】

【特許文献】

10

【0005】

【特許文献1】特開平8-314744号公報

【発明の概要】

【発明が解決しようとする課題】

【0006】

これまで専用ハードウェアによって各計算機が接続されていた従来のフォールトトレラントシステムの構成では、各計算機において同一時刻に同一のプログラムを実行することは可能であった。

一方、近年ではネットワークの高速化、及び計算機の汎用化に伴い、専用ハードウェアではなく、一般的なネットワークを利用して複数計算機を接続する技術が発達するとともに、計算機システムが処理する情報量の拡大に伴い、各処理ステップを複数のタスクとして分解し、マルチタスク実行可能OSが複数のタスクを切り換えながら実行するニーズがある。

20

【0007】

しかしながら、マルチタスクを実行可能なOSが搭載されたマルチプロセッサを有する計算機において、他の計算機と同期をとるべき同期タスクと、他の計算機との同期が必要ない非同期タスクとを、どのようにマルチプロセッサに実行させるかについては、考えられてこなかった。

【0008】

本発明は上記のような課題に鑑みてなされたものであり、マルチプロセッサ環境を有する複数の計算機をネットワークで接続した計算機システムにおいて、計算機間でのタスク同期を必要とする同期タスクと、タスク同期を必要としない非同期タスクのプロセッサへの割当についての技術を提供する。

30

【課題を解決するための手段】

【0009】

計算機は、他の計算機と同期をとってタスクを処理する同期プロセッサと、他の計算機と同期をとらずにタスクを処理する非同期プロセッサと、同期プロセッサに割り当てられたタスクの状態を管理する同期タスク管理テーブルと、非同期プロセッサに割り当てられたタスクの状態を管理する非同期タスク管理テーブルと、同期プロセッサ及び非同期プロセッサにタスクを割り当てるディスパッチャとを有する。なお、当該計算機と他の計算機は同期タスク管理テーブルの内容を、タスク制御用共有メモリ区画を介して相互に通信して交換している。

40

【0010】

ディスパッチャは、タスクが他の計算機と同期をとって処理すべきタスクか否かを判断し、当該判断結果に基づいて、他の計算機と同期をとりながら処理すべき同期タスクは同期プロセッサに割当て、他の計算機との同期が不要な非同期タスクは非同期プロセッサに割当て、同期タスク又は非同期タスクのいずれとしてでも処理可能な同期・非同期タスクについて同期プロセッサ又は非同期プロセッサのどちらに割り当てるかを判定して、該当プロセッサに割当て、同期プロセッサが実行している実行中同期タスクを、他のタスクに切り替える切替ポイントを検出した場合に、実行中同期タスクの状態を待機中に変更し、

50

当該計算機の同期タスク管理テーブルの内容と他の計算機の同期タスク管理テーブルの内容とに基づいて、当該計算機の同期タスク管理テーブルに実行可能タスクとして登録されている複数の同期タスクの内の一を選択して同期プロセッサに実行させる。

【0011】

その他本願が開示する課題やその解決方法については、発明の実施形態の欄及び図面により明らかにされる。

【発明の効果】

【0012】

本発明によれば、マルチプロセッサ環境を有する複数の計算機をネットワークで接続した計算機システムにおいて、計算機間でのタスク同期を実現することができる。

10

【図面の簡単な説明】

【0013】

【図1】本実施形態の計算機システムの構成例を示す図である。

【図2】本実施形態の計算機の構成例を示す図である。

【図3】本実施形態のタスク管理テーブルの構成例を示す図である。

【図4】本実施形態の非同期用のプロセッサに対して行われるディスパッチを説明する図である。

【図5】本実施形態の非同期用のプロセッサに対して行われるアン・ディスパッチを説明する図である。

【図6】本実施形態の同期用のプロセッサに対するディスパッチを説明する図である。

20

【図7】同時に複数のタスクが実行可能になった場合のタスク選定処理を説明する図である。

【図8】本実施形態の同期用のプロセッサに対するアン・ディスパッチを説明する図である。

【図9】本実施形態の切替システムコールが呼び出された場合のアン・ディスパッチを説明する図である。

【図10】本実施形態の切替システムコールが呼び出された場合のディスパッチを説明する図である。

【図11】断片化したメモリに連続領域のメモリ領域を確保する処理を行うタスクの一例を示す図である。

30

【図12】タスクを割り当てるプロセッサを切り替えて、断片化したメモリに連続領域のメモリ領域を確保する処理を行うタスクの一例を示す図である。

【図13】本実施形態の同期用のプロセッサから非同期用のプロセッサにタスクを割り当てるプロセッサを変更する処理を説明する図である。

【図14】本実施形態の非同期用のプロセッサから同期用のプロセッサにタスクを割り当てるプロセッサを変更する処理を説明する図である。

【図15】本実施形態のタスクプログラムがタスクの同期または非同期を指示する場合の処理を説明する図である。

【図16】本実施形態の計算機間で行うタスク管理テーブルの交換がタイムアウトした場合の処理を説明する図である。

40

【図17】本実施形態のタイムアウトを起こした回数を考慮するタイムアウト処理を説明する図である。

【図18】Voterを含む計算機システムの構成例を示す図である。

【図19】Voterにおける多数決処理に参加する計算機を変更する処理を説明する図である。

【図20】計算機における同期実行をおこなう計算機を変更する処理を説明する図である。

【図21】ソフトウェアにエラーが生じた場合の処理を説明する図である。

【図22】ハードウェアにエラーが生じた場合の処理を説明する図である。

【図23】プロセス間通信の有無を含めてタスクプログラムの同期を行う処理を説明する

50

図である。

【図 2 4】 あいこ状態を考慮した多数決によるタスクの選定処理を説明する図である。

【図 2 5】 本実施形態の非同期用のプロセッサにて同期タスクの実行をシリアルライズするためのタスク管理テーブルの構成例を示す図である。

【図 2 6】 本実施形態の非同期用のプロセッサにて同期タスクの実行をシリアルライズするための処理を説明する図である。

【図 2 7】 本実施形態の同期用のプロセッサから非同期用のプロセッサに同期タスクを割り当てる際のシリアルライズ処理を説明する図である。

【図 2 8】 本実施形態の非同期用のプロセッサから同期用のプロセッサに同期タスクを割り当てる際のシリアルライズ完了処理を説明する図である。

10

【図 2 9】 本実施形態のプロセッサ個別にメモリ回収を行うための構成例を示す図である。

【図 3 0】 本実施形態のプロセッサ個別にメモリ回収を行う処理を示す図である。

【図 3 1】 本実施形態のシステムにおける、同期タスクおよび非同期タスクの定義テーブルの配置を示す図である。

【図 3 2】 本実施形態の同期タスクおよび非同期タスクの定義テーブルの内容を説明する図である。

【図 3 3】 本実施形態のタスクを管理する T C B の内容を説明する図である。

【図 3 4】 本実施形態の同期用のプロセッサから非同期用のプロセッサに同期タスクを割り当てる際の処理を説明する図である。

20

【図 3 5】 本実施形態の非同期用のプロセッサから同期用のプロセッサに同期タスクを割り当てる際の処理を説明する図である。

【図 3 6】 本実施形態のタスク起動処理を説明する図である。

【発明を実施するための形態】

【実施例 1】

【0 0 1 4】

——システム構成——

以下に実施形態について図面を用いて詳細に説明する。図 1 は、本実施形態の計算機システム 1 0 の構成図である。図 2 は計算機 1 0 0 A 内の構成を詳細に示す図である。

【0 0 1 5】

30

計算機システム 1 0 は、マルチプロセッサ環境を有する複数の計算機をネットワークで接続した計算機システムであって、計算機間でのタスク同期を行い、フォールトトレラントシステムを構成するコンピュータシステムである。

計算機システム 1 0 は、少なくとも 2 台の計算機 1 0 0 A と計算機 1 0 0 B がネットワーク 4 0 0 および 5 0 0 によって接続されている。計算機 1 0 0 A と計算機 1 0 0 B は同一の構成を有しており、以下では、詳細な構造については計算機 1 0 0 A についてのみ説明する。

【0 0 1 6】

計算機 1 0 0 A は、2 つのプロセッサ 1 8 0 A (1 8 0 A - 1 および 1 8 0 A - 2) 、タイマー 1 7 0 A 、通信インターフェース 1 5 0 A および 1 6 0 A 、ならびにメモリ 1 1 0 A を備え、これらはバス 1 9 0 A にて接続されている。

40

本実施形態では、プロセッサ 1 8 0 A - 1 は、計算機 1 0 0 B との間でタスクを同期するタスク同期用として用い、プロセッサ 1 8 0 A - 2 は、計算機 1 0 0 B とは独立して動作するタスク非同期用として用いるものとする。なお、タスク非同期用のプロセッサ 1 8 0 A - 2 の用途が一般的である。

【0 0 1 7】

メモリ 1 1 0 A は、2 つのタスクプログラム 1 2 0 (1 2 0 A および 1 2 0 B) 、オペレーティングシステム 1 4 0 A 、ならびに、計算機 1 0 0 B との間でのタスク同期に必要な情報を送受するためのタスク制御用共有メモリ区画 1 3 0 A などを備えている。なお、タスクプログラム 1 2 0 には、オペレーティングシステムを実現するために必要なシステ

50

ムプログラムと、それ以外のユーザプログラムとを含む。本実施形態では、システムプログラムは不図示とし、ユーザプログラムのみをタスクプログラム120として図示することとする。

【0018】

オペレーティングシステム140Aは、複数のタスクを実行することのできるマルチタスク実行可能なオペレーティングシステムである。なお、オペレーティングシステム140Aは、プロセッサ180Aがメモリ110Aに記憶されているシステムプログラムを実行することにより実現される。すなわち、オペレーティングシステム140Aも、タスクが実行されることにより実現される。オペレーティングシステム140Aは、タスクプログラム120の実行に係る処理をタスクに分解し、各タスクをプロセッサ180A-1または180A-2に割り当てるディスパッチャ141A、タスク同期を行うためにタスク間メッセージを考慮したタスク同期を行うためのシグナル管理テーブル112A・タスク間メッセージ管理テーブル143A、プロセッサ180A-1および180A-2に割り当てたタスクに関する情報（「タスク制御ブロック；Task Control Block」という。以下、「TCB」と表記する。）を記憶する2つのタスク管理テーブル143（143A-1および143A-2）、同期計算機管理テーブル111Aを備える。タスク管理テーブル143A-1は、プロセッサ180A-1に割り当てられたタスクのTCBを記憶し、タスク管理テーブル143A-2は、プロセッサ180A-2に割り当てられたタスクのTCBを記憶する。

10

【0019】

オペレーティングシステム140Aは、計算機100A内のリソースを複数のタスクに割り当てるために、プリエンプションを行う技術を提供している。プリエンプションとは、実行中のタスクを一時的に打ち切り、他のタスクにリソースを使用させることである。

20

【0020】

後述するように、各タスクには優先度が設定可能となっており、オペレーティングシステム140Aは、優先度に応じて、タスクの実行順序を決定する。実行優先度の高いタスクが存在する場合は、実行優先度が低いタスクはプリエンプションにより実行権限を一時的に打ち切られる。オペレーティングシステム140Aは、タスクごとに定められる所定の時間タスクを実行した場合にも、そのタスクの実行を打ち切る。

【0021】

タスクプログラム120には、複数の命令が含まれる。図2には、タスクプログラム120Aの構成例が図示されている。タスクプログラム120Aには、制御命令列A120-1、システムコール120-2、および制御命令列B120-3が含まれている。ここでシステムコール120-2とは、タスクからオペレーティングシステム140Aに対する命令である。システムコール120-2としては、例えば、ディスクへのアクセス命令や、タイマー170Aからの時刻の取得命令などがある。オペレーティングシステム140Aは、タスクから呼び出されるシステムコールに応じて、例えば、ディスクへのアクセスや、タイマー170Aからの時刻の取得を行うなどの各種の処理を行う。

30

【0022】

近年計算機は、より高速に処理を実行するためにキャッシュ機能を提供している。プログラムが使用する情報がキャッシュにヒットする場合、計算機が保有するメモリから情報を取得する必要がないため、高速に動作することが可能となる。本実施形態の計算機システム10では、クロック同期などを行うことなく計算機100Aと計算機100Bとは独立に動作するため、例えば、計算機100Aにおいてキャッシュに情報が格納された場合にも計算機100Bではキャッシュに情報が格納されないという状況が発生し得る。しかし、このような状況においても、後述するように、計算機100A及び100Bがタスクをディスパッチする際に、同じタスクをディスパッチするように情報を交換することにより同期を行うことが可能となっている。

40

【0023】

プロセッサ180A-1は、プログラムカウンタ180A-1a、汎用レジスタ180

50

A-1b、および制御レジスタ180A-1cを備えている。なお、プロセッサ180A-2も同様の構成である。

プログラムカウンタ180A-1aは、プロセッサ180A-1が次に実行すべき命令が格納されているメモリ110A上のアドレスを指し示すレジスタである。プロセッサ180A-1がメモリ110Aから命令を読み込む度に、読み取った命令の分だけ値が増加する。

【0024】

汎用レジスタ180A-1bは、各種の値を記憶するレジスタである。汎用レジスタ180A-1bは、プロセッサ180A-1が行う演算処理などに用いられる。

制御レジスタ180A-1cは、各種のフラグをリセットし、あるいはプロセッサ180A-1の動作モードを特定のモード（例えば、低消費電力モード）にするように指示するためのレジスタである。

【0025】

図3は、タスク管理テーブル143の構成例を示す図である。計算機100Aで実行されるタスクは、生成されてから消滅するまでの間、実行中の状態、実行可能であるがまだ実行されていない状態（実行可能状態）、条件成就や入力などを待機している状態（待機状態）の間でその状態が遷移する。タスク管理テーブル143には、実行可能状態であるタスクに係るTCB146が格納される実行可能タスクキュー144、待機状態であるタスクに係るTCB146が格納される待機状態タスクキュー145、および実行中のタスクのTCB146を格納する実行中タスク147が含まれる。一般的には実行可能タスクキュー144には、優先度の高い順にTCB146が格納され、先頭から順にタスクがディスパッチされることにより、優先度の高い順にタスクがディスパッチされるようになっている。

【0026】

TCB146には、ポインタ146-0が含まれる。実行可能タスクキュー144および待機状態タスクキュー145は、キューの先頭のTCB146を示すポインタを含み、実行可能タスクキュー144および待機状態タスクキュー145に格納されているTCB146のポインタ146-0には、キューにおける次のTCB146を示すポインタが格納される。キューの最後尾のTCB146のポインタ146-0にはnullが設定される。

【0027】

また、TCB146には、タスクID146-1、プログラムカウンタ146-2、汎用レジスタ146-3、制御レジスタ146-4、割り込み可否146-5、プリエンブトタイプ146-6が含まれる。

【0028】

タスクは、タスクID146-1により識別される。プログラムカウンタ146-2は、プロセッサ180A-1のプログラムカウンタ180A-1aに設定される値である。

【0029】

汎用レジスタ146-3は、プロセッサ180A-1の汎用レジスタ180A-1bに設定される値である。制御レジスタ146-4は、プロセッサ180A-1の制御レジスタ180A-1cに設定される値である。ディスパッチャ141Aは、プロセッサ180A-1において実行中のタスクの実行を打ち切って他のタスクに切り替える場合に、プログラムカウンタ180A-1a、汎用レジスタ180A-1b、および制御レジスタ180A-1cの内容をTCB146に設定して、タスクが実行可能状態である場合には実行可能タスクキュー144に追加し、タスクが待機状態である場合には、待機状態タスクキュー145に追加し、実行可能タスクキュー144から1つTCB146を読み出して、TCB146のプログラムカウンタ146-2、汎用レジスタ146-3、および制御レジスタ146-4の内容をプロセッサ180A-1に設定することにより、タスクの切り替えを行う。

【0030】

本実施形態では、ディスパッチャ141Aが、プロセッサ180Aのカウンタやレジスタの内容をTCB146に設定して実行中のタスクを打ち切ることを「アン・ディスパッチ」といい、TCB146に基づいてプロセッサ180Aに新たなタスクを実行させることを「ディスパッチ」ということとする。

【0031】

タスク制御用共有メモリ区画130は、各計算機100間で分散共有されるメモリである。タスク制御用共有メモリ区画130は、タスクの同期処理を行う頻度よりも高い頻度でOS140により計算機100間で内容の同期が行われているものとする。後述するように、タスク制御用共有メモリ区画130には、実行可能タスクキュー144に格納されているTCB146がコピーされる。これによりディスパッチャ141は、各計算機100における実行可能タスクキュー144に格納されているTCB146を交換することが可能になっている。

10

【0032】

同期計算機管理テーブル111は、同期をとろうとする計算機100を示す識別情報に対応付けて、当該計算機100と同期がとれているか否かを示すフラグやステータス情報を管理する。

シグナル管理テーブル112には、タスク間通信のためにシグナルが発生するごとに、発生したシグナルに関する情報がキューとして登録される。

【0033】

以下、本実施形態における計算機制御方法の実際手順について図に基づき説明する。以下で説明する計算機制御方法に対応する各種動作は、計算機システム10を構成する各装置のRAM(Random Access Memory)に読み出して実行するプログラムによって実現される。そして、このプログラムは、以下に説明される各種の動作を行うためのコードから構成されている。

20

【0034】

——非同期ディスパッチ——

図4は、非同期用のプロセッサ180A-2に対して行われるディスパッチ（待ち状態からディスパッチ）を説明する図である。なお、一般的なプロセッサに対するディスパッチの処理も、図4に示すようなものになる。

ディスパッチャ141Aは、タスク管理テーブル143A-2の実行可能タスクキュー144より、先頭のTCB146Aを選定する(S701)。すなわち、ディスパッチャ141Aは、実行可能タスクキュー144のポインタが示すTCB146Aを選定する。

30

【0035】

ディスパッチャ141Aは、選定したTCB146Aより、プログラムカウンタ情報146-2の情報をプログラムカウンタ180A-2aに格納し、汎用レジスタ情報146-3の情報を汎用レジスタ180A-2bに格納し、制御レジスタの情報146-4を制御レジスタ180A-2cに格納する(S702)。ディスパッチャ141Aは、実行中タスク147のポインタが示すTCB146Eを待機状態タスクキュー145の最後尾に追加する。すなわち、ディスパッチャ141は、待機状態タスクキュー145のポインタが示すTCB146Cから、TCB146のポインタ146-0がnullになる最後尾のTCB146Dまで辿り、最後尾のTCB146Dのポインタ146-0にTCB146Eを示すポインタを設定し、TCB146Eのポインタ146-0にnullを設定する。ディスパッチャ141Aは、実行可能タスクキュー144の先頭から選定したTCB146Aを実行中タスク147として設定する。すなわち、ディスパッチャ141Aは、TCB146Aを示すポインタを実行中タスク147のポインタに設定する(S703)。

40

【0036】

また、ディスパッチャ141Aは、実行可能タスクキュー144のポインタに、TCB146Aの次に位置するTCB146Bを示すポインタを設定する。ディスパッチャ141Aは、プロセッサ180A-2に実行を指示する(S704)。

50

このようにして、プロセッサ180A-2におけるタスクの切り替えが行われる。

【0037】

——非同期アン・ディスパッチ——

図5は、非同期用プロセッサ180A-2に対して行われるアン・ディスパッチ（ディスパッチから待ち状態）を説明する図である。なお、一般的なプロセッサに対するアン・ディスパッチの処理も、図5に示すようなものになる。

ディスパッチャ141Aは、プロセッサ180A-2に実行中断を指示し（S705）、プロセッサ180A-2の状態を、実行中タスク147のポインタが示すTCB146Aに退避する。すなわち、ディスパッチャ141Aは、プログラムカウンタ180A-2aの情報をプログラムカウンタ情報146-2に格納し、汎用レジスタ180A-2bの情報を汎用レジスタ情報146-3に格納し、制御レジスタ180A-2cの情報を制御レジスタ情報146-4に格納する（S706）。

これにより、プロセッサ180A-2の状態がTCB146Aに退避され、アン・ディスパッチ処理が終了する。

【0038】

——同期ディスパッチ——

図6は、ディスパッチャ141によるオペレーティングシステム140内での同期用のプロセッサ180A-1に対する同期ディスパッチを説明する図である。図6に示す処理は、オペレーティングシステム140内タスク同期制御（待ち状態からディスパッチ）である。一般的なディスパッチとの違いはステップS714の前に同期処理が行われることである。なお、図6のS714は、図4のS702～S704に対応する。

【0039】

オペレーティングシステム140Aにおいてディスパッチャ141Aが起動されると（S710）、ディスパッチャ141Aは、計算機100Bのディスパッチャ141Bとの間で、タスク管理テーブル143A-1内の実行可能タスクキュー144（以下、「実行可能タスクキュー144A」という。）の内容と、タスク管理テーブル143B-1内の実行可能タスクキュー144（以下、「実行可能タスクキュー144B」という。）の内容とを通信インターフェース160および150を介して交換し、各同期計算機100Aと100Bの制御用共有メモリ区画130A、130Bに各々格納する（S712）。ディスパッチャ141Aは、実行可能タスクキュー144Aに登録されているTCB146と、実行可能タスクキュー144Bに登録されているTCB146との両方の情報を制御用共有メモリ区画130より取得し、優先度を含めて一致化しているかを確認し、次にプロセッサ180A-1に割り当てるべきタスクを選定する。（S713）。

【0040】

図7は、プロセッサ180A-1に割り当てるべき（ディスパッチするべき）タスクが存在する場合のタスクの選定処理の一例を説明する図である。各計算機100において同一優先度の実行可能タスクキュー141に同時に複数のタスクが存在する場合を想定している。

【0041】

ディスパッチャ141Aは、実行可能タスクキュー144の内容をタスク制御用共有メモリ区画130Aに書き込む。上述したように、タスク制御用共有メモリ区画130Aは、他の計算機100と共有されているので、各計算機100の実行可能タスクキュー144の内容がタスク制御用共有メモリ区画130Aに書き込まれることになる。

【0042】

ディスパッチャ141Aは、同期計算機管理テーブル111Aを参照し、同期を取る計算機100（以下、「同期計算機」ともいう。）がn台であることを確認する（S7131）。ディスパッチャ141Aは、同期を取る計算機100のそれぞれからタスク制御用共有メモリ区画130Aに書き込まれているTCB146のタスクID146-1のうち、タスク制御用共有メモリ区画130Aにn個存在するものがあるか否かを判断する（S7132）。ディスパッチャ141Aは、n個存在するタスクIDのうち、識別子が最も

若い同期計算機 100 の実行可能タスクキュー 144 の最も先頭に近いものに対応するタスクをディスパッチするタスクとして選定する (S7133)。

【0043】

ディスパッチャ 141A は、上記 n 個存在するタスク ID 146-1 がいない場合には、タスク ID 146-1 が n 個タスク制御用共有メモリ区画 130A に書き込まれるまで、所定のタイムアウト時間待機する。タイムアウト時間が経過しても、n 個のタスク ID 146-1 が書き込まれない場合には、いずれかの計算機 100 の同期が外れた事が考えられる。そこで、タイムアウト時間が経過した場合、ディスパッチャ 141A は、n を 1 つ減算し (S7134)、n が 2 より多い場合には (S7135:NO)、n 個のタスク ID 146-1 がタスク制御用共有メモリ区画 130A に書き込まれているか否かを判断する (S7136)。ディスパッチャ 141A は、上記 n 個のタスク ID 146-1 が存在する場合 (S7136:YES)、当該タスク ID 146-1 のうち識別子が最も若い計算機 100 の実行可能タスクキュー 144 の最も先頭に近いものを選択し、選択したタスク ID 146-1 が実行可能タスクキュー 144 に含まれていない計算機 100 を、同期外れを起こした計算機 100 として、同期計算機管理テーブル 111A より除外する (S7137)。ディスパッチャ 141A は、S7133 に進み、同期が外れた計算機 100 を除外した上で、上記選択したタスク ID に対応するタスクをディスパッチするタスクとして選定する。ディスパッチャ 141A は、n 個のタスク ID 146-1 がタスク制御用共有メモリ区画 130A に書き込まれていない場合には (S7136)、S7134 に戻り、さらに n を 1 減算する。

【0044】

n が 2 以下、すなわち同期している計算機 100 が 2 台以下となり、多数決が取れない場合、もしくは、n が 1 で同期する計算機 100 が 1 台しか存在しなくなった場合には (S7135:YES)、ディスパッチャ 141A は、同期計算機 100 のうち識別子の最も若い計算機 100 を同期計算機管理テーブル 111A に残し、その他の計算機 100 を同期計算機管理テーブル 111A から除外する (S7138)。ディスパッチャ 141A は、S7133 に進み、同期計算機管理テーブル 111A に残された計算機 100 の実行可能タスクキュー 144 の最も先頭のタスク ID 146-1 に対応するタスクをディスパッチするタスクとして選定する。

【0045】

以上のようにして、同期計算機管理テーブル 111 を使用し、同期計算機情報を更新しながら多数決によりタスクの選定が行われる。

【0046】

図 7 の処理によりディスパッチすべきタスクが選定されると、ディスパッチャ 141A は、図 4 の S702~S704 と同様に、選定したタスクの TCB 146 の内容をプロセッサ 180A-1 に設定して、プロセッサ 180A-1 に選定したタスクを実行させる (S714)。

【0047】

以上のようにして、全ての計算機 100 において、実行可能タスクキュー 144 を交換した上で、全ての実行可能タスクキュー 144 に基づいてディスパッチすべきタスクを決定するので、全ての計算機 100 において同じタスクを次に割り当てるタスクに決定することができる。これにより、実行可能タスクキュー 144 に登録されている TCB 146 が計算機 100 によって異なる場合でも、全ての計算機 100 において同じタスクがディスパッチされるようにすることができる。よって、クロック同期等を行うことなく、タスクの同期を行うことができる。

【0048】

——同期アン・ディスパッチ——

図 8 は、ディスパッチャ 141 によるオペレーティングシステム 140 内での同期用のプロセッサ 180A-1 に対するアン・ディスパッチを説明する図である。図 8 に示す処理は、オペレーティングシステム 140 内タスク同期制御 (実行状態から待ち状態) であ

る。

ディスパッチャ 141A は、オペレーティングシステム 140A からタスク 120B を待機させる旨の指示を受けると (S715)、プロセッサ 180A-1 が実行中であるタスク 120B の情報を回収する (S716)。すなわち、ディスパッチャ 141A は、図 5 の S705 および S706 と同様に、プロセッサ 180A-1 に実行を中断するように指示し、プロセッサ 180A-1 のプログラムカウンタ 180A-1a、汎用レジスタ 180A-1b および制御レジスタ 180A-1c の内容を、タスク管理テーブル 143A-1 に含まれる実行中タスク 147 のポインタが示す TCB 146E に設定する。ディスパッチャ 141A は、TCB 146E を、タスク管理テーブル 143A-1 の待ち状態タスクキュー 145 に移動する (S717)。

10

【0049】

——同期アン・ディスパッチ——

上記のように、ディスパッチおよびアン・ディスパッチのタイミング (切替可能ポイント) は、オペレーティングシステム 140A のディスパッチャ 141A が決定するが、タスクプログラム 120 がそのタイミングを指示するようにすることもできる。この場合、所定のシステムコールを呼び出す API (Application Program Interface) を規定しておくようにする。以下、ディスパッチやアン・ディスパッチのタイミングを決定するための API に規定されたシステムコールを「切替システムコール」という。また、図 2 のタスクプログラム 120A に含まれるシステムコール 120-2 は、切替システムコールであるものとする。

20

【0050】

図 9 は、切替システムコールが呼び出された場合のアン・ディスパッチを説明する図である。図 9 に示す処理は、ディスパッチャ 141 によるタスク内での API を介したタスク同期制御 (実行状態から待ち状態) である。システムコール 120-2 がアン・ディスパッチを指示する切替システムコールであり、タスクがシステムコール 120-2 を呼び出すと (S720)、ディスパッチャ 141A は、図 8 の S716、S717 の処理を行う。

【0051】

——API による同期ディスパッチ——

図 10 は、切替システムコールが呼び出された場合のディスパッチを説明する図である。図 10 に示す処理は、ディスパッチャ 141 によるタスク内での API を介したタスク同期制御 (待ち状態から実行状態) である。システムコール 120-2 がディスパッチを指示する切替システムコールであり、タスクが、切替先としてタスク 120A を指定してシステムコール 120-2 を呼び出した場合 (S730)、ディスパッチャ 141A は、タスク管理テーブル 143A-1 内の待ち状態タスクキュー 145 に格納されている、タスク 120A に係る TCB 146 を、実行可能タスクキュー 144 に移動する (S732)。ディスパッチャ 141A は、計算機 100B のディスパッチャ 141B との間で、タスク管理テーブル 143A-1 およびタスク管理テーブル 143B-1 内の実行可能タスクキュー 144 に格納されている TCB 146A、146B を通信インターフェース 160 および 150 を介して交換する (S733)。ディスパッチャ 141A は、タスク管理テーブル 143A-1 の実行可能タスクキュー 144 に登録されている TCB 146 と、タスク管理テーブル 143B-1 の実行可能タスクキュー 144 に登録されている TCB 146 との両方に基づいて、次にプロセッサ 180A-1 に割り当てるべきタスクを選定する (S734)。ディスパッチャ 141A は、図 4 の S702-S704 と同様に、選定したタスクの TCB 146 の内容をプロセッサ 180A-1 に設定して、プロセッサ 180A-1 に選定したタスクを実行させる (S735)。

30

40

【0052】

——同期・非同期の切り替え——

オペレーティングシステム 140A は、例えば、プログラムの起動時にユーザから指定されたオプションにより、当該プログラムの実行に係るタスクを他の計算機 100 と同期

50

すべきか否かを決定することができる。ディスパッチャ 141A は、同期すべきタスクは同期用のプロセッサ 180A-1 に割り当て、非同期でよいタスクは非同期用のプロセッサ 180A-2 に割り当てる。

【0053】

オペレーティングシステム 140A は、タスクにおいて実行される処理に応じてタスクを同期すべきか否かを決定することもできる。例えば、タスクから同期を指示する所定のシステムコール（以下、「同期システムコール」という。）が呼び出された場合、ディスパッチャ 141A は、そのタスクを同期用のプロセッサ 180A-1 に割り当て、タスクから非同期を指示する所定のシステムコール（以下、「非同期システムコール」という。）が呼び出された場合には、そのタスクを非同期用のプロセッサ 180A-2 に割り当てるようにすることができる。

10

【0054】

図 11 は、断片化したメモリに連続領域のメモリ領域を確保する処理を行うタスクの一例を示す図である。すなわち、連続したメモリ領域が確保できるか否かが判断され（S12001）、連続したメモリ領域が確保できない場合には（S12001：NO）、メモリの回収タスクが起動され（S12002）、連続したメモリ領域を確保できる場合には（S12001：YES）、当該連続したメモリ領域を確保することになる（S12003）。各計算機 100 におけるメモリの使用状況等は異なることが多いので、図 11 のタスクが同期用のプロセッサ 180A-1 にディスパッチされてしまうと、同期を取ることが難しくなる場合がある。そこで、連続したメモリ領域の確保を行う処理など、各計算機 100 の状態に依存する処理（以下、「環境依存処理」という。）を行うタスクを同期用のプロセッサ 180A-1 に割り当てる場合には、図 12 に示すように、環境依存処理を行う前にタスクを非同期とし、環境依存処理の後に処理を行う場合にはタスクを同期するようにすることができる。すなわち、タスクが非同期システムコールを呼び出すと（S12004）、ディスパッチャ 141A は、当該タスクを同期用プロセッサ 180A-1 から非同期用プロセッサ 180A-2 に割り当てを変更することで、非同期に上記図 11 と同様にして連続したメモリ領域が確保されることになる。なお、プロセッサ 180A 間のタスクの割り当ての変更処理の詳細については後述する。次に、タスクが同期システムコールを呼び出すと（S12005）、ディスパッチャ 141A は、非同期用プロセッサ 180A-2 から同期用のプロセッサ 180A-1 に当該タスクの割り当てを変更し、これによりそれ以後のタスクは同期をとって実行されることになる。

20

30

【0055】

このように、タスクにおいて実行される処理に応じて同期及び非同期を切り替えるようにすることで、同期に適さない計算機 100 それぞれの資源や環境に応じた処理を行う場合には、非同期でタスクを実行し、それ以外の場合には同期させてタスクを実行するようにすることもできる。

【0056】

――同期タスクの非同期への変更――

図 13 は、同期用のプロセッサ 180A-1 から非同期用のプロセッサ 180A-2 にタスクを割り当てるプロセッサを変更する、オペレーティングシステム 140 内における同期タスクの非同期化処理を説明する図である。図 13 に示す処理は、オペレーティングシステム 140 内タスク同期制御（実行状態から別コアでの非同期実行状態）である。

40

【0057】

オペレーティングシステム 140A が、ディスパッチャ 141A にタスク 120B を非同期にするように指示した場合（S740）、ディスパッチャ 141A は、同期用のプロセッサ 180A-1 が実行しているタスク 120B の情報を回収する（S741）。すなわち、ディスパッチャ 141A は、図 5 の S705、S706 と同様に、プロセッサ 180A-1 に実行を中断するように指示し、プロセッサ 180A-1 のプログラムカウンタ 180A-1a、汎用レジスタ 180A-1b および制御レジスタ 180A-1c の内容を、タスク管理テーブル 180A-1 に含まれる実行中タスク 147 のポインタが示す T

50

C B 1 4 6 E に設定する。ディスパッチャ 1 4 1 A は、プロセッサ 1 8 0 A - 1 に対応するタスク管理テーブル 1 4 3 A - 1 からタスク 1 2 0 B に対応する T C B 1 4 6 E を削除し (S 7 4 2) 、非同期用のプロセッサ 1 8 0 A - 2 に対応するタスク管理テーブル 1 4 3 A - 2 の実行可能タスクキュー 1 4 4 に、タスク 1 2 0 B についての T C B 1 4 6 E を登録する (S 7 4 3) 。これにより、ディスパッチャ 1 4 1 A は、タスク管理テーブル 1 4 3 A - 2 の実行可能タスクキュー 1 4 4 に登録された T C B 1 4 6 E に基づいて、図 4 に示すディスパッチを行い、プロセッサ 1 8 0 A - 2 においてタスク 1 2 0 B が実行される (S 7 4 4) 。

【 0 0 5 8 】

——非同期タスクの同期への変更——

10

図 1 4 は、非同期用のプロセッサ 1 8 0 A - 2 から同期用のプロセッサ 1 8 0 A - 1 にタスクを割り当てるプロセッサを変更する処理を説明する図である。図 1 4 に示す処理は、オペレーティングシステム 1 4 0 内タスク同期制御 (別コアでの非同期実行状態から同期コアでのディスパッチ) である。

オペレーティングシステム 1 4 0 A が、ディスパッチャ 1 4 1 A にタスク 1 2 0 B を同期するように指示した場合 (S 7 5 0) 、ディスパッチャ 1 4 1 A は、プロセッサ 1 8 0 A - 2 が実行しているタスク 1 2 0 B の情報を回収する (S 7 5 1) 。すなわち、ディスパッチャ 1 4 1 A は、図 5 の S 7 0 5 、 S 7 0 6 と同様に、プロセッサ 1 8 0 A - 2 に実行を中断するように指示し、プロセッサ 1 8 0 A - 2 のプログラムカウンタ 1 8 0 A - 2 a 、汎用レジスタ 1 8 0 A - 2 b および制御レジスタ 1 8 0 A - 2 c の内容を、タスク管理テーブル 1 8 0 A - 2 に含まれる実行中タスク 1 4 7 のポインタが示す T C B 1 4 6 E に設定する。ディスパッチャ 1 4 1 A は、プロセッサ 1 8 0 A - 2 に対応するタスク管理テーブル 1 4 3 A - 2 からタスク 1 2 0 B に対応する T C B 1 4 6 E を削除し (S 7 5 2) 、同期用のプロセッサ 1 8 0 A - 1 に対応するタスク管理テーブル 1 4 3 A - 1 の実行可能タスクキュー 1 4 4 に、タスク 1 2 0 B についての T C B 1 4 6 E を登録する (S 7 5 3) 。これにより、ディスパッチャ 1 4 1 A は、図 6 の S 7 1 2 ~ 7 1 4 と同様に計算機 1 0 0 B のディスパッチャ 1 4 1 B との間で、タスク管理テーブル 1 4 3 A - 1 およびタスク管理テーブル 1 4 3 B - 1 内の実行可能タスクキュー 1 4 4 に格納されている T C B 1 4 6 A 、 1 4 6 B を通信インターフェース 1 6 0 および 1 5 0 を介して交換し (S 7 5 4) 、タスク管理テーブル 1 4 3 A - 1 の実行可能タスクキュー 1 4 4 に登録されている T C B 1 4 6 と、タスク管理テーブル 1 4 3 B - 1 の実行可能タスクキュー 1 4 4 に登録されている T C B 1 4 6 との両方に基づいて、次にプロセッサ 1 8 0 A - 1 に割り当てるべきタスクを選定する (S 7 5 5) 。これにより、ディスパッチャ 1 4 1 A は、タスク管理テーブル 1 4 3 A - 1 の実行可能タスクキュー 1 4 4 に登録された T C B 1 4 6 E に基づいて、図 4 に示すディスパッチを行い、プロセッサ 1 8 0 A - 1 においてタスク 1 2 0 B が実行される (S 7 5 6) 。

20

30

【 0 0 5 9 】

——API によるタスクの同期非同期の変更——

タスクプログラム 1 2 0 がタスクの同期または非同期を指示するようにすることもできる。この場合、タスクの同期または非同期を指示するシステムコール (以下、「同期システムコール」または「非同期システムコール」という。) を呼び出す API を規定しておくようにする。図 1 5 はこの場合における、同期プロセッサと非同期プロセッサとの間でのタスクの移動によるタスク内での同期タスクの非同期化・同期化処理を説明する図である。タスク 1 2 0 B から非同期システムコールが呼び出された場合 (S 7 6 0) 、ディスパッチャ 1 4 1 A は、図 1 3 の S 7 4 1 ~ S 7 4 4 の処理を行う。また、タスク 1 2 0 B から同期システムコールが呼び出された場合には (S 7 7 0) 、ディスパッチャ 1 4 1 A は、図 1 4 の S 7 5 1 ~ S 7 5 6 の処理を行う。

40

これにより、タスクプログラムから任意に同期または非同期への変更を指定することが可能となる。

【 0 0 6 0 】

50

――タイムアウト処理――

図16は、例えば図6のS712などにおいて、計算機100Aと計算機100Bとの間で行うタスク管理テーブル143の交換がタイムアウトした場合の処理を説明する図である。図16に示す処理は、計算機100の同期外れ判定（時間での判定）による同期タスク待ちのタイムアウト処理である。

ディスパッチャ141Aは、処理を開始した時刻を取得して（S781）、タスク管理テーブル143A-1およびタスク管理テーブル143B-1内の実行可能タスクキュー144に格納されているTCB146A、146Bを通信インターフェース160および150を介して交換し、各同期計算機100A、100Bのタスク制御用共有メモリ区画130A、130Bに各々のTCB146を格納する（S782）。 10

【0061】

タスク管理テーブル143A-1の実行可能タスクキュー144に登録されているTCB146と、計算機100Bから取得したタスク管理テーブル143B-1の実行可能タスクキュー144に格納されているTCB146とが一致していれば（S783：YES）、図6のS713、S714と同様に、ディスパッチャ141Aは、タスク管理テーブル143A-1の実行可能タスクキュー144に登録されているTCB146と、タスク管理テーブル143B-1の実行可能タスクキュー144に登録されているTCB146との両方に基づいて、次にプロセッサ180A-1に割り当てべきタスクを選定する（S785）。ディスパッチャ141Aは、図4のS702～S704と同様に、選定したタスクのTCB146の内容をプロセッサ180A-1に設定して、プロセッサ180A-1に選定したタスクを実行させる（S786）。 20

【0062】

一方、TCB146が一致しなければ（S783：NO）、ディスパッチャ141Aは、上記S781で取得した時刻から現在時刻までの時間が所定の時間を越えたか否かにより、タイムアウトを起こしたか否かを判定する（S784）。タイムアウトを起こしていなければ（S784：NO）、ディスパッチャ141Aは、S782からの処理を繰り返す。タイムアウトを起こしていれば（S784：YES）、ディスパッチャ141Aは、実行可能タスクを多数決により決定する（S787）。すなわち、各計算機の実行可能タスクキュー144について、登録されているTCB146に基づいてタスクを選定していき、最も多く選定されたタスクを選定するようにする。最も多く選定されたタスクが複数存在する場合には、例えば、予め計算機の優先度を決めておき、優先度の高い計算機について選定されたタスクを優先して選定するようにする。ディスパッチャ141Aは、タスク制御用共有メモリ区画130Aを参照し、多数決により選定されなかった計算機100を特定し、特定した計算機100を示す識別情報に対応づけて、当該計算機100と同期がとれていない旨を示す情報を同期計算機管理テーブル111Aに登録する（S788）。ディスパッチャ141Aは、図4のS702～S704と同様に、選定したタスクのTCB146の内容をプロセッサ180A-1に設定して、プロセッサ180A-1に選定したタスクを実行させる（S789）。 30

【0063】

――回数を考慮したタイムアウト処理――

図17は、上述した図16のタイムアウト処理においてタイムアウトを起こした回数を考慮する処理を説明する図である。図17に示す処理は、計算機100の同期外れ判定（回数での判定）による同期タスク待ちのタイムアウト処理である。図17のS790～S797の処理は、図16のS781～S788と同様である。ディスパッチャ141Aは、S797の後、カウンタをインクリメントして、カウンタが所定の設定回数を越えたか否かを判定し（S798）、カウンタが設定回数を越えていなければ（S798：NO）、観察実行とする（S799）。観察実行とは、例えば計算機識別子が多数決において高い優先度を持っていたとしても、優先度が最低として扱う方式である。つまり、図7のS7138において、識別子の若い計算機が観察実行状態であれば、S7138では識別子の年取った方が残る。カウンタが設定回数を越えていれば（S798：YES）、計算機 40 50

を再起動する（S800）。

【0064】

――Voter870を含むシステム構成――

図18は、計算機システム10が、計算機100Aからの出力と計算機100Bからの出力とが一致することを検証するVoter870（多数決機）を含む構成を示す図である。

【0065】

Voter870は、計算機システム10を構成する複数の計算機100からの出力が一致していればそのまま出力し、異なる場合には、多数決により最も一致した数の多いものを出力する多数決処理を行う。Voter870は、メモリ810、プロセッサ820、通信インターフェース860を備えており、これらがバス830により接続されている。メモリ810は、オペレーティングシステム840を備え、オペレーティングシステム840は、同期計算機管理テーブル850を備えている。同期計算機管理テーブル850は計算機100が備える同期計算機管理テーブル111と同様の構成である。同期計算機管理テーブル850には、多数決処理に参加する計算機100が登録されており、オペレーティングシステム840は、同期計算機管理テーブル850に登録されている計算機100からの出力のみについて多数決処理を行う。

【0066】

以下、Voter870を備える計算機システム10における処理について説明する。

【0067】

――計算機100からVoter870への通知処理――

図19は、Voter870における多数決処理に参加する計算機100を変更する処理を説明する図である。図19に示す処理は、同期計算機管理情報の送受信（計算機100からVoter870）であり、タスク同期外れした計算機100をVoter870へ通知し、出力結果の多数決に参加できないようにするものである。計算機100Aのディスパッチャ141Aは、上述した図16のS788や図17のS797において、多数決により選定されなかった計算機100をVoter870に通知し（S841）、Voter870のオペレーティングシステム840は、同期計算機管理テーブル850から、通知された計算機100を削除する（S842）。これにより、タイムアウトした計算機100はVoter870による多数決処理に参加しないようにすることができる。

【0068】

――Voter870から計算機100への通知処理――

図20は、計算機100における多数決処理に参加する計算機100を変更する処理を説明する図である。図20に示す処理は、同期計算機管理情報の送受信（Voter870から計算機100）であり、Voter870において出力結果の多数決負けを起こした計算機100を他の計算機100に通知し、同期をとって同期計算機管理テーブル111から当該計算機100を除外するものである。Voter870のオペレーティングシステム840は、上記多数決処理において選定されなかった計算機100を、全ての計算機100に通知する（S843）。計算機100のディスパッチャ141は、Voter870から通知された計算機100を同期計算機管理テーブルから削除する（S844）。

【0069】

――ソフトウェアエラーの処理――

図21は、ソフトウェアにエラーが生じた場合の処理を説明する図である。図21に示す処理は、オペレーティングシステム140内タスク同期制御（待ち状態からディスパッチ）におけるエラーハンドリング処理（ソフトウェアエラー）であり、一般的なディスパッチとの違いはS714の前に同期処理が発生することである。ディスパッチャ141AによりS711～S714の処理が行われた後、タスクにエラーが発生した場合（S900）、ディスパッチャ141Aは、タスクがエラーを出力していれば（S901：YES）、エラー処理を行うプログラムを構成するタスクについてのTCB146を、実行可能

10

20

30

40

50

タスクキュー１４４の先頭に移動し（Ｓ９０２）、エラーを出力していなければ（Ｓ９０１：ＮＯ）、図６のＳ７１１と同様に実行可能状態にあるタスクを１つ選定する（Ｓ９０３）。ディスパッチャ１４１Ａは、計算機１００Ｂのディスパッチャ１４１Ｂとの間で、タスク管理テーブル１４３Ａ－１およびタスク管理テーブル１４３Ｂ－１内の実行可能タスクキュー１４４に格納されているＴＣＢ１４６Ａ、１４６Ｂを通信インターフェース１６０および１５０を介して交換し（Ｓ９０４）、タスク管理テーブル１４３Ａ－１の実行可能タスクキュー１４４に登録されているＴＣＢ１４６と、タスク管理テーブル１４３Ｂ－１の実行可能タスクキュー１４４に登録されているＴＣＢ１４６との両方に基づいて、次にプロセッサ１８０Ａ－１に割り当てるべきタスクを選定する（Ｓ９０５）。ディスパッチャ１４１は、図４のＳ７０２～Ｓ７０４と同様に、選定したタスクのＴＣＢ１４６の内容をプロセッサ１８０Ａ－１に設定して、プロセッサ１８０Ａ－１に選定したタスクを実行させる（Ｓ９０６）。

10

【００７０】

——ハードウェアエラーの処理——

図２２は、ハードウェアにエラーが生じた場合の処理を説明する図である。図２２に示す処理は、オペレーティングシステム１４０内タスク同期制御（待ち状態からディスパッチ）におけるエラーハンドリング処理（ハードウェアエラー）であり、一般的なディスパッチとの違いはＳ７１４の前に同期処理が行われることである。ディスパッチャ１４１ＡによりＳ７１１～Ｓ７１４の処理が行われた後、ハードウェアにエラーが発生したことを、監視タスクやオペレーティングシステム１４０Ａが検知した場合（Ｓ９１０）、ディスパッチャ１４１Ａは、エラーが存在していれば（Ｓ９１１：ＹＥＳ）、他の計算機１００Ｂにエラーの内容を示すエラー情報の複製を送信し（Ｓ９１２）、エラー処理を行うプログラムを構成するタスクについてのＴＣＢ１４６を、実行可能タスクキュー１４４の先頭に移動する（Ｓ９１３）。エラーが存在しなければ（Ｓ９１１：ＮＯ）、ディスパッチャ１４１Ａは、図６のＳ７１１と同様に実行可能状態にあるタスクを１つ選定する（Ｓ９１４）。ディスパッチャ１４１Ａは、計算機１００Ｂのディスパッチャ１４１Ｂとの間で、タスク管理テーブル１４３Ａ－１およびタスク管理テーブル１４３Ｂ－１内の実行可能タスクキュー１４４に格納されているＴＣＢ１４６Ａ、１４６Ｂを通信インターフェース１６０および１５０を介して交換し（Ｓ９１５）、タスク管理テーブル１４３Ａ－１の実行可能タスクキュー１４４に登録されているＴＣＢ１４６と、タスク管理テーブル１４３Ｂ－１の実行可能タスクキュー１４４に登録されているＴＣＢ１４６との両方に基づいて、次にプロセッサ１８０Ａ－１に割り当てるべきタスクを選定する（Ｓ９１６）。ディスパッチャ１４１は、図４のＳ７０２～Ｓ７０４と同様に、選定したタスクのＴＣＢ１４６の内容をプロセッサ１８０Ａ－１に設定して、プロセッサ１８０Ａ－１に選定したタスクを実行させる（Ｓ９１７）。

20

30

【００７１】

——タスク間通信を考慮したタスク同期処理——

図２３は、計算機１００において同期タスクが実行されている場合に、同期タスク間通信をプロセス間通信の有無を含めてタスクプログラムの同期を行う処理を説明する図である。

40

【００７２】

オペレーティングシステム１４０Ａがタスクからタスク間通信の相手先への通信の依頼を受けると（Ｓ９２０）、オペレーティングシステム１４０Ａは、タスク間通信の内容をタスク間メッセージ管理テーブル１４２Ａ・シグナル管理テーブル１１２Ａにキューとしてつないで管理し（Ｓ９２１）、計算機１００Ｂのディスパッチャ１４１Ｂとの間で、タスク管理テーブル１４３Ａ－１およびタスク管理テーブル１４３Ｂ－１内の実行可能タスクキュー１４４に格納されているＴＣＢ１４６Ａ、１４６Ｂと、シグナル管理テーブル１１２Ａ及び１１２Ｂとタスク間メッセージテーブル１４２Ａ及び１４２Ｂに記憶されている情報とを通信インターフェース１６０および１５０を介して交換し、タスク制御用共有メモリ区画１３０Ａ～１３０Ｂに各々格納する（Ｓ９２３）。ディスパッチャ１４１Ａは

50

、タスク管理テーブル143A-1の実行可能タスクキュー144に登録されているTCB146と、タスク管理テーブル143B-1の実行可能タスクキュー144に登録されているTCB146とが、同期タスクへの通信有無を含めて一致しているか否かを確認する。すなわち、ディスパッチャ141Aは、全ての実行可能キュー144A及び144Bに登録されている全てのTCB146について、タスクID146-1に対応する情報をシグナル管理テーブル112A又は112Bに登録されている場合、タスクID146-1に対応する情報をシグナル管理テーブル112A及び112Bのそれぞれから取得し、取得した情報の内容が一致していなければ、当該TCB146を実行可能キュー144Aから削除して待機状態タスクキュー145に追加する(S924)。ディスパッチャ141Aは、上記図7の処理と同様にして、次にプロセッサ180A-1に割り当てるべきタスクを選定する(S925)。ディスパッチャ141Aは、図4のS702-S704と同様に、選定したタスクのTCB146の内容をプロセッサ180A-1に設定して、プロセッサ180A-1に選定したタスクを実行させる(S926)。ディスパッチャ141Aは、ディスパッチしたタスクに係るTCB146のタスクID146-1に対応する情報がシグナル管理テーブル112Aに登録されている場合には、通信内容をタスク通信の相手先に通達する(S927)。このようにして、次に実行するタスクを決定する際に、シグナル管理テーブル145に登録されているタスク間通信の内容が計算機100Aと計算機100Bとの間で一致することを確認することにより、タスク間通信が行われている場合でも、タスク間通信の内容に影響を与えることなく、計算機100Aと計算機100Bとの間でディスパッチの同期を行うことが可能となる。

10

20

【0073】

以上説明したように、例えばリアルタイムタスク等、優先度の高いプロセスの実行を行うためにプリエンプションを実行している場合に、現行のマルチタスク実行可能なオペレーティングシステムでは、タイマーに基づいた判断によりプリエンプションを行うため、プリエンプションが行われたタスク処理の進捗が計算機間で変わり、各計算機間での同期が取れなくなるところ、本実施形態によれば、複数の計算機100の間でタスク管理テーブル143を交換し、複数のタスク管理テーブル143に基づいて次にディスパッチするタスクを決定することができる。すなわち、ある計算機100において実行可能となっているタスクであっても、他の計算機100において実行可能となっていなければ、ディスパッチの対象として選定されないようにすることができる。これにより、複数の計算機100の間でディスパッチするタスクの同期を取ることが可能となる。

30

【0074】

また、キャッシュに該当情報がヒットしたか否かにより、システムコール(例:ハードディスクドライブへのデータ読み取り指示等)が発行されたりされなかったりすることが考えられるが、この場合にも、従来のマルチタスク実行可能なオペレーティングシステムではタスク処理スピードの変化が生じて同期が取れなくなるところ、本実施形態によれば、全ての計算機100において実行可能となったタスクのみをディスパッチすることができるので、タスク処理スピードが変化した場合であっても、複数の計算機100間でディスパッチの同期を取ることが可能となる。

40

【0075】

また、マルチコアやマルチプロセッサ環境において、本実施形態の計算機システム10によれば、タスク内の処理フローの一部においてタスクを割り当てるコアやプロセッサを変更することができる。

【0076】

また、本実施形態の計算機システム10によれば、タスクの同期をとる処理と、タスクの同期を外して非同期に処理する方式とを交互に制御し、その制御の境目で複数コアを移動させることができる。

【0077】

また、本実施形態の計算機システム10によれば、タスクが同期を外れるために他のコアへ移動するためのAPIやシステムコールをタスクに提供することができる。

50

【0078】

また、本実施形態の計算機システム10によれば、他のコアへ移動し同期が外れたタスクが、再び同期を取るために元のコアへ戻ってくるためのAPIやシステムコールをタスクに提供することができる。

【0079】

また、本実施形態の計算機システム10によれば、他計算機との同期を意識して、障害等が発生した場合に、他の計算機にエラーを通知して、エラー処理を行うことができる。

【0080】

したがって、高信頼システムを実現できる。

【0081】

また、本実施形態の計算機システム10によれば、Voterによる精査の結果を受け、タスク同期の制御にフィードバックすることができる。

【0082】

なお、これまで示した計算機システム10をなす計算機100A、100Bにおける各機能部、Voter870における各機能部等はハードウェアとして実現してもよい、各装置におけるメモリやHDD(Hard Disk Drive)などの適宜な記憶部に格納したプログラムとして実現するとしてもよい。この場合、各装置のCPUなど制御部がプログラム実行に合わせて記憶部より該当プログラムを読み出して、これを実行することとなる。

【0083】

また、本実施形態の計算機100Aは、2つのプロセッサ180A-1および180A-2を備えるものとしたが、3つ以上のプロセッサを備えるようにしてもよい。

【0084】

また、計算機100Aは、複数のコアを有する1つのプロセッサのみを備えるようにしてもよい。この場合、オペレーティングシステム140Aは、各コアについてTCBを管理し、タスクのディスパッチを行うものとする。

【0085】

また、本実施形態では、実行可能タスクキュー144には優先度の高い順にTCB146が格納されているものとしたが、TCB146に優先度を記憶するようにして、実行可能タスクキュー144には、キュー以外の形態で、順番を問わずにTCB146を管理するようにしてもよい。この場合、ディスパッチャ141Aは、図4のS701において、実行可能タスクキュー144に格納されているTCB146のうち、優先度の最も高いものを選択するようにする。

【0086】

また、本実施形態では、Voter870による多数決処理は、出力が一致する計算機100を複数のグループに分けた場合に、所属する計算機100の数が同じであるグループが複数存在する状態(あいこ状態)を考慮していなかったが、あいこ状態を考慮するようにしてもよい。図24は、あいこ状態を考慮する場合のVoter870による多数決処理を説明する図である。この場合、各計算機100には、重複しない番号(計算機No)を割り当てておくものとする。また、図24の処理では、計算機100から出力される内容を重複なく抽出して比較対象値とし、各計算機100からの出力をその比較対象値に対する投票として、多数決処理を説明する。Voter870は、次に投票数の多い比較対象値を選び(S8701)、選んだ比較対象値に対する投票数と同じ投票数の他の比較対象値が存在する場合(あいこ状態)には(S8702:YES)、計算機Noが最も小さい計算機100が投票した比較対象値を出力するようにし(S8703)、あいこ状態でない場合には(S8702:NO)、選んだ比較対象値を出力する(S8704)。これにより、あいこ状態を考慮した多数決処理により、出力を決定することができる。なお、計算機No以外にも、例えば、計算機100に優先度を割り当てておき、優先度に応じて、Voter870が出力を選択するようにしてもよい。

【0087】

以上の実施形態によって、マルチプロセッサ環境を有する複数の計算機をネットワークで接続した計算機システムにおいて、計算機間でのタスク同期が可能となる。即ち、一般的なネットワークを利用して複数計算機を接続する方式において、従来のマルチタスク動作可能なOSは、タスクスケジュールの中で他の計算機の挙動を意識するように作られてはいなかった。したがって、従来技術では、例えば、プリエンプションが行われたタスク処理の進捗が計算機間で異なってしまう状況や、キャッシュに該当情報がヒットしたか否かにより、計算機によってシステムコール（例：ハードディスクドライブへのデータ読み取り指示等）が発行されたりされなかったりという違いが生じる状況などに対処することが出来ていない。これにより、タスク開始や終了のタイミングを各計算機間で合わせる事ができず、ひいては計算機間での同期は取れなくなってしまうていた。つまり従来技術における構成や処理内容を、一般的なネットワークにて複数計算機を接続する方式に単に適用したとしても、フォールトトレラントシステムに求められる要求をそもそも満たせなくなっていた。本実施例ではこのような従来技術の課題を解決し、各々がマルチプロセッサを有する複数の計算機間でのタスク同期が可能となる。

10

【実施例2】

【0088】

次に他の実施形態について説明する。本実施形態では、実施例1において、複数の同期タスクについて一部の処理が同期用プロセッサから非同期用プロセッサに割り当てが変更され、一時的に非同期用プロセッサにて実行されることになった場合を想定している。この時、これら複数の同期タスクについて複数計算機間での同期を一時的にとることができなくなる。そこで、各計算機において非同期用プロセッサで実行される同期タスクの実行順序を管理し、非同期用プロセッサに同期タスクが割り当てられた順序に従って、同期タスクが非同期用プロセッサで実行されるよう制御する方法を説明する。尚、本実施例において特に説明しない限り、システムの構成、処理方法は実施例1と同様である。

20

【0089】

ー非同期実行用プロセッサでの同期タスクのディスパッチー

まず、本実施形態では、図3に示すタスク管理テーブル143に代えて図25に示すタスク管理テーブル143-2が使用される。図25に示すタスク管理テーブル143-2は、図3に加え同期タスク順序管理キュー148があり、同期実行用プロセッサから同期タスクが非同期実行用プロセッサへ割当変更された順序で当該同期タスクのTCBへのリンク194がつながっている。各リンク194は、同期用プロセッサのタスク管理テーブル143から非同期プロセッサ用のタスク管理テーブル143に移し換えられ、実行可能タスクキュー144につなげられた同期タスクのTCBへのポインタを少なくとも保持している。更に図25には図示していないが、各TCB146にはタスクの種類146-7が追加されている。このタスクの種類146-7はタスクが同期タスクであるか非同期タスクであることを示す情報が格納されている。

30

更に本実施例では図4に示す一般的・タスク非同期ディスパッチ処理に代えて図26に示す一般的・タスク非同期ディスパッチ処理が実行される。図26に示す処理には、図4に示す処理に更にS705～S707が追加されており、図26に示す処理を実行すると、非同期用のプロセッサ180A-2に対して、同期タスクの実行順序を管理しつつディスパッチを実行することができる。

40

【0090】

ディスパッチャ141Aは、タスク管理テーブル143-2の実行可能タスクキュー144より、先頭のTCB146を選定する（S701）。すなわち、ディスパッチャ141Aは、実行可能タスクキュー144のポインタが示すTCB146Aを選定する。ディスパッチャ141Aは、実行可能タスクキュー先頭のTCB146が同期タスクか確認する（S705）。この確認は、TCB146のタスクの種類146-7を確認することによって行われる。このTCB146が非同期タスクであればS702を実行する。TCB146が同期タスクであれば（S705：YES）、ディスパッチャ141Aは、実行可能タスクキュー先頭のTCB146が同期タスク順序管理キュー148の先頭のタスクと

50

同一か確認する（S706）。すなわち、この確認により同期タスクが非同期実行用プロセッサで処理される順番を制御することができる。もし、実行可能タスクキュー先頭のTCB146が同期タスク順序管理キュー148の先頭のタスクと同一であれば（S706：YES）、S702を実行する。

【0091】

実行可能タスクキュー先頭のTCB146が同期タスク順序管理キュー148の先頭のタスクと同一でない場合は（S705：NO）、実行順序待機中なので実行可能タスクキューの先頭のTCBを、実行可能タスクキューの最後尾に繋ぎ直す（S707）。繋ぎなおした後に、再びS701に戻る。

【0092】

S705またはS706にて実行可能タスクが選定されると、ディスパッチャ141Aは、選定したTCB146Aより、プログラムカウンタ情報146-2の情報をプログラムカウンタ180A-2aに格納し、汎用レジスタ情報146-3の情報を汎用レジスタ180A-2bに格納し、制御レジスタの情報146-4を制御レジスタ180A-2cに格納する（S702）。ディスパッチャ141Aは、実行中タスク147のポインタが示すTCB146Eを待機状態タスクキュー145の最後尾に追加する。すなわち、ディスパッチャ141Aは、待機状態タスクキュー145のポインタが示すTCB146Cから、TCB146のポインタ146-0がnullになる最後尾のTCB146Dまで辿り、最後尾のTCB146Dのポインタ146-0にTCB146Eを示すポインタを設定し、TCB146Eのポインタ146-0にnullを設定する。ディスパッチャ141Aは、実行可能タスクキュー144の先頭から選定したTCB146Aを実行中タスク147として設定する。すなわち、ディスパッチャ141Aは、TCB146Aを示すポインタを実行中タスク147のポインタに設定する（S703）。また、ディスパッチャ141Aは、実行可能タスクキュー144のポインタに、TCB146Aの次に位置するTCB146Bを示すポインタを設定する。ディスパッチャ141Aは、プロセッサ180A-2に実行を指示する（S704）。

この処理ステップにより、複数の同期タスクが非同期実行用プロセッサにて実行され、複数計算機間での同期を一時的にとらない場合においても、計算機内での実行順序を規定することができる。

【0093】

——同期タスクの非同期への変更——

更に本実施形態では、図13に示す処理に代えて図27に示す処理を用いて、同期用のプロセッサ180A-1から非同期用のプロセッサ180A-2にタスクを割り当てるプロセッサを変更する。図27においては図13におけるS744に変えてS745が実行される結果、同期タスクの実行をシリアライズするための登録処理が行われる。

オペレーティングシステム140Aが、ディスパッチャ141Aにタスク120Bを非同期にするように指示した場合（S740）、ディスパッチャ141Aは、同期用のプロセッサ180A-1が実行しているタスク120Bの情報を回収する（S741）。すなわち、ディスパッチャ141Aは、図5のS705、S706と同様に、プロセッサ180A-1に実行を中断するように指示し、プロセッサ180A-1のプログラムカウンタ180A-1a、汎用レジスタ180A-1bおよび制御レジスタ180A-1cの内容を、タスク管理テーブル180A-1に含まれる実行中タスク147のポインタが示すTCB146Eに設定する。ディスパッチャ141Aは、プロセッサ180A-1に対応するタスク管理テーブル143A-1からタスク120Bに対応するTCB146Eを削除し（S742）、非同期用のプロセッサ180A-2に対応するタスク管理テーブル143A-2の実行可能タスクキュー144に、タスク120BについてのTCB146Eを登録する（S743）。非同期実行用プロセッサ180A-2用タスク管理テーブル内の同期タスク順序管理キュー148の最後尾にタスク120Bへのリンク149を登録する（S745）。

【0094】

これにより、ディスパッチャ 141A は、タスク管理テーブル 143A-2 の実行可能タスクキュー 144 に登録された TCB 146E に基づいて、図 26 に示すディスパッチを行い、プロセッサ 180A-2 においてタスク 120B が実行される。

【0095】

――非同期タスクの同期への変更――

更に本実施形態では、図 14 に示す処理に代えて図 28 に示す処理を用いて、非同期用のプロセッサ 180A-2 から同期用のプロセッサ 180A-1 にタスクを割り当てるプロセッサを変更する。図 28 においては、図 14 の処理に更に S757 が追加されることによって、同期タスクのシリアライズが行われる。

オペレーティングシステム 140A が、ディスパッチャ 141A にタスク 120B を同期するように指示した場合 (S750)、ディスパッチャ 141A は、プロセッサ 180A-2 が実行しているタスク 120B の情報を回収する (S751)。すなわち、ディスパッチャ 141A は、図 5 の S705、S706 と同様に、プロセッサ 180A-2 に実行を中断するように指示し、プロセッサ 180A-2 のプログラムカウンタ 180A-2a、汎用レジスタ 180A-2b および制御レジスタ 180A-2c の内容を、タスク管理テーブル 180A-2 に含まれる実行中タスク 147 のポインタが示す TCB 146E に設定する。ディスパッチャ 141A は、プロセッサ 180A-2 に対応するタスク管理テーブル 143A-2 からタスク 120B に対応する TCB 146E を削除し (S752)、非同期実行用プロセッサ 180A-2 用タスク管理テーブル内の同期タスク順序管理キュー 148 からタスク 120B へのリンクを削除する (S752)。

【0096】

その後、同期用のプロセッサ 180A-1 に対応するタスク管理テーブル 143A-1 の実行可能タスクキュー 144 に、タスク 120B についての TCB 146E を登録する (S753)。これにより、ディスパッチャ 141A は、図 6 の S712~714 と同様に計算機 100B のディスパッチャ 141B との間で、タスク管理テーブル 143A-1 およびタスク管理テーブル 143B-1 内の実行可能タスクキュー 144 に格納されている TCB 146A-146B を通信インターフェース 160 および 150 を介して交換し (S754)、タスク管理テーブル 143A-1 の実行可能タスクキュー 144 に登録されている TCB 146 と、タスク管理テーブル 143B-1 の実行可能タスクキュー 144 に登録されている TCB 146 との両方に基づいて、次にプロセッサ 180A-1 に割り当てるべきタスクを選定する (S755)。これにより、ディスパッチャ 141A は、タスク管理テーブル 143A-1 の実行可能タスクキュー 144 に登録された TCB 146E に基づいて、図 26 に示すディスパッチを行い、プロセッサ 180A-1 においてタスク 120B が実行される (S756)。

【実施例 3】

【0097】

更に他の実施形態を示す。本実施形態は、実施例 1 又は実施例 2 において、メモリ回収タスクの同期用プロセッサによる実行を抑制するための構成を示す。尚、本実施例において特に説明しない限り、システムの構成、処理は実施例 1 または実施例 2 と同様である。

【0098】

――プロセッサ毎メモリ回収機能を含むシステム構成――

本実施形態は、図 1 に示すシステム構成に代えて図 29 に示すシステム構成を有する。図 29 に示すシステムは、図 1 のシステムに加えて、メモリ回収タスクプログラム 120C を有し、またオペレーティングシステム 140A 内にプロセッサ毎メモリ回収閾値管理テーブル 113 を含む。

【0099】

――プロセッサ毎メモリ回収機能の処理――

図 30 は、メモリ取得時にメモリ取得前にフォアグラウンドで行われるメモリ回収処理とバックグラウンドにて行われるメモリ回収処理における、メモリ回収タスクの処理の一例である。この処理はメモリ回収タスクプログラム 120C が実行されることによって行

10

20

30

40

50

われる。

メモリ回収タスクはタスク管理テーブル143よりメモリ回収タスクを実行するCPUを検知する(S121)。すなわち、プロセッサ毎に設定されているメモリ回収閾値を取得するため、まずはメモリ回収タスクを実行する可能性のあるプロセッサの識別を行う。次に検知されたプロセッサ毎に、当該プロセッサについてのメモリ回収閾値以下に現在のメモリ空き容量が達し、メモリを圧迫しているかを確認する(S122)。メモリ空き容量がメモリ回収閾値以下に達していた場合、当該プロセッサにメモリ回収を実行させる(S123)。非同期用プロセッサは同期用プロセッサよりも高い頻度でメモリ回収タスクが稼動する様に、非同期用プロセッサのメモリ回収閾値は同期用プロセッサのメモリ回収閾値より厳しく(大きく)してある。この結果、図11や図12に示した様な同期タスクがメモリ取得の一環でフォアグラウンドにてメモリ回収タスクを動作させることを、抑止することができる。

10

【0100】

尚、メモリ回収タスクは一例である。その他のタスクについても、同期用プロセッサと非同期用プロセッサとでタスクの実行契機の判断基準となる閾値を異なる値にしておくことによって、同期用プロセッサではなく非同期用プロセッサに当該タスクをなるべく実行させ、あるいは逆に非同期用プロセッサではなく同期用プロセッサに当該タスクをなるべく実行させるよう構成することができる。

【実施例4】

【0101】

20

更に他の実施形態を示す。本実施形態は、実施例1乃至実施例3において、タスクを同期用プロセッサ又は非同期用プロセッサに割り当てる際の処理の一例を示す。尚、本実施例において特に説明しない限り、システムの構成、処理は実施例1乃至実施例3と同様である。

【0102】

――システム構成――

――同期および非同期タスクの定義テーブルの配置――

図31に示すシステムは、図29に示すシステムに加えて、記録媒体600および同期タスク非同期タスク定義テーブル610を含む。同期タスク非同期タスク定義テーブル610は、図31に示す記録媒体600の他、外部の記録媒体に置くことも可能である。

30

【0103】

――本実施形態の同期タスクおよび非同期タスクの定義テーブルの内容――

図32は、計算機100Aの同期タスク非同期タスク定義テーブル610の内容の一例である。同期タスク非同期タスク定義テーブル610には、タスク名610-1と、実行時のタスクID610-2、タスクの種別610-3、同期用プロセッサ項目610-4、および非同期用プロセッサ項目610-5が格納されている。タスクの種別には、「同期」「非同期」「同期・非同期」がある。「同期」はタスクを同期タスクとして起動する場合、「非同期」はタスクを非同期タスクとして起動する場合、「同期・非同期」はタスクを同期タスクとしてでも非同期タスクとしてでも起動可能な場合をあらわす。同期プロセッサ項目610-4は、タスクの種別610-3が「同期」又は「同期・非同期」の場合に当該タスクを実行する同期用プロセッサを指定する情報である。非同期プロセッサ項目610-5は、タスクの種別が「同期」である場合には当該同期タスクが一時的な非同期処理を行う場合にその非同期処理を実行するべき非同期用プロセッサを指定する情報であり、タスクの種別が「非同期」または「同期・非同期」である場合に当該タスクを実行すべき非同期用プロセッサを指定する情報である。

40

【0104】

本実施例では2つのプロセッサを有する計算機での例を示すが、2つ以上のプロセッサを有する計算機であっても同期タスク非同期タスク定義テーブル610は利用可能である。

【0105】

50

――タスクを管理するTCBの内容――

図31に示すシステムが有するタスク管理テーブル143は、図3に示すタスク管理テーブルとは異なる構成を有する。具体的にはタスク管理テーブル143内のTCB146の構成が異なる。そこで図33にTCB146の構成の一例を示す。図33に示すTCB146は、図3に示すTCBに加え、同期タスクや非同期タスクなどタスクの種類を示すタスクの種類146-7、同期タスクを実行する同期用プロセッサを指定する同期実行時プロセッサ識別子146-8、同期タスクの中で一時的に非同期処理を行う場合にその非同期処理を実行する非同期用プロセッサ、もしくは非同期タスクを実行する非同期用プロセッサを指定する非同期実行時プロセッサ識別子146-9の項目を有する。

【0106】

10

――同期タスクの非同期への変更――

図34は、同期用のプロセッサ180A-1から非同期用のプロセッサ180A-2にタスクを割り当てるプロセッサを変更する際、非同期用プロセッサを定義情報から判定し非同期処理を行う方法を説明する図である。

オペレーティングシステム140Aが、ディスパッチャ141Aにタスク120Bを非同期にするように指示した場合(S740)、ディスパッチャ141Aは、同期用のプロセッサ180A-1が実行しているタスク120Bの情報を回収する(S741)。すなわち、ディスパッチャ141Aは、図5のS705、S706と同様に、プロセッサ180A-1に実行を中断するように指示し、プロセッサ180A-1のプログラムカウンタ180A-1a、汎用レジスタ180A-1bおよび制御レジスタ180A-1cの内容を、タスク管理テーブル180A-1に含まれる実行中タスク147のポインタが示すTCB146Eに設定する。ディスパッチャ141Aは、プロセッサ180A-1に対応するタスク管理テーブル143A-1からタスク120Bに対応するTCB146Eを削除する(S742)。

20

【0107】

ディスパッチャ141Aは、タスク120BのTCB146Eの非同期実行時プロセッサ識別子146-9より、タスク120Bが実行される非同期用プロセッサ識別子を取得する。すなわち、非同期用プロセッサ180A-2であるという情報を取得する(S1000)。非同期用のプロセッサ180A-2に対応するタスク管理テーブル143A-2の実行可能タスクキュー144に、タスク120BについてのTCB146Eを登録する(S743)。非同期実行用プロセッサ180A-2用タスク管理テーブル内の同期タスク順序管理キュー148の最後尾にタスク120Bへのリンクを登録する(S745)。

30

【0108】

これにより、ディスパッチャ141Aは、タスク管理テーブル143A-2の実行可能タスクキュー144に登録されたTCB146Eに基づいて、図26に示すディスパッチを行い、プロセッサ180A-2においてタスク120Bが実行される。

【0109】

――非同期タスクの同期への変更――

図35は、非同期用のプロセッサ180A-2から同期用のプロセッサ180A-1にタスクを割り当てるプロセッサを変更する際のシリアライズ完了処理を説明する図である。

40

オペレーティングシステム140Aが、ディスパッチャ141Aにタスク120Bを同期するように指示した場合(S750)、ディスパッチャ141Aは、プロセッサ180A-2が実行しているタスク120Bの情報を回収する(S751)。すなわち、ディスパッチャ141Aは、図5のS705、S706と同様に、プロセッサ180A-2に実行を中断するように指示し、プロセッサ180A-2のプログラムカウンタ180A-2a、汎用レジスタ180A-2bおよび制御レジスタ180A-2cの内容を、タスク管理テーブル180A-2に含まれる実行中タスク147のポインタが示すTCB146Eに設定する。ディスパッチャ141Aは、プロセッサ180A-2に対応するタスク管理テーブル143A-2からタスク120Bに対応するTCB146Eを削除し(S752

50

）、非同期実行用プロセッサ180A-2用タスク管理テーブル内の同期タスク順序管理キュー148からタスク120Bへのリンクを削除する（S756）。

【0110】

ディスパッチャ141Aは、タスク120BのTCB146Eの同期実行プロセッサ識別子146-8より、タスク120Bが実行される同期用プロセッサ識別子を取得する。すなわち、180A-1であるという情報を取得する（S1001）。その後、同期用のプロセッサ180A-1に対応するタスク管理テーブル143A-1の実行可能タスクキュー144に、タスク120BについてのTCB146Eを登録する（S753）。これにより、ディスパッチャ141Aは、図6のS712～714と同様に計算機100Bのディスパッチャ141Bとの間で、タスク管理テーブル143A-1およびタスク管理テーブル143B-1内の実行可能タスクキュー144に格納されているTCB146A-146Bを通信インターフェース160および150を介して交換し（S754）、タスク管理テーブル143A-1の実行可能タスクキュー144に登録されているTCB146と、タスク管理テーブル143B-1の実行可能タスクキュー144に登録されているTCB146との両方に基づいて、次にプロセッサ180A-1に割り当てるべきタスクを選定する（S755）。

10

【0111】

これにより、ディスパッチャ141Aは、タスク管理テーブル143A-1の実行可能タスクキュー144に登録されたTCB146Eに基づいて、図26に示すディスパッチを行い、プロセッサ180A-1においてタスク120Bが実行される。

20

【0112】

――タスクの起動処理――

図36は、同期タスク・非同期タスク、またはメモリ回収タスクのように必要に応じて同期タスクからも非同期タスクからも起動されるタスクの起動プロセスを説明する図である。図36に示す処理をディスパッチャ141によって実行される。

【0113】

タスクが起動される場合、ディスパッチャ141は記録媒体600内の同期タスク非同期タスク管理テーブル610より、起動を指定されたタスク名610-1を選出する（S1002）。該当したタスク名610-1の行からタスクの種類610-3を参照し、該当タスクの種類が同期か、非同期か、または同期、非同期両方かを判定する。

30

【0114】

同期タスクと判断された場合、当該タスクを610-4に示されるプロセッサで実行するために、メモリ確保などの起動処理を実行する（S1006）。更に、該当タスク用TCB146を生成し、TCB146内のタスクの種類146-7に該当する同期タスク非同期タスク定義テーブル610のタスクの種類610-3の情報を書き込み、TCB146内の同期実行時プロセッサ識別子146-8に該当する同期タスク非同期タスク定義テーブル610の同期プロセッサ610-4を書き込み、TCB146内の非同期実行時プロセッサ識別子146-9に該当する同期タスク非同期タスク定義テーブル610の非同期プロセッサ610-9の情報を書き込む（S1007）。ディスパッチャ141は当該タスクのTCB146を同期実行時プロセッサ識別子146-8にて示されるプロセッサのタスク管理テーブル143A-1へ登録する（S1008）

40

ステップS1003で非同期タスクと判断された場合（S1003：YES）、当該タスクを610-5に示されるプロセッサで実行するために、メモリ確保などの起動処理を実行する（S1010）。該当タスク用TCB146を生成し、TCB146内のタスクの種類146-7に該当する同期タスク非同期タスク定義テーブル610のタスクの種類610-3の情報を書き込み、TCB146内の同期実行時プロセッサ識別子146-8に該当する同期タスク非同期タスク定義テーブル610の同期プロセッサ610-4を書き込み、TCB146内の非同期実行時プロセッサ識別子146-9に該当する同期タスク非同期タスク定義テーブル610の非同期プロセッサ610-5の情報を書き込む（S1007）。ディスパッチャ141は当該タスクのTCB146を146-9にて示され

50

るプロセッサのタスク管理テーブル143A-2へ登録する(S1011)。

【0115】

ステップS1003で「同期・非同期」タスクと判断された場合、該当タスクの起動を指示した親タスクの実行プロセッサ識別子が、同期タスク非同期タスク管理テーブル610の同期プロセッサ610-4に含まれるか確認する(S1004)。同期プロセッサ610-4に含まれる場合はS1006へ進み(S1004:YES)、同期タスクとしての起動ステップを踏む。同期プロセッサ610-4に該当しない場合は(S1004:NO)、該当タスクの起動を指示した親タスクの実行プロセッサ識別子が同期タスク非同期タスク管理テーブル610の非同期プロセッサ610-5に含まれるか確認する(S1005)。非同期プロセッサ610-5に含まれる場合はS1010に進み(S1005:YES)、非同期タスクとしての起動ステップを踏む。その他該当タスクの起動を指示した親タスクの実行プロセッサ識別子が同期タスク非同期タスク管理テーブル610の非同期プロセッサ610-5に含まれない場合は想定外の起動指示としてタスクの起動を中止し起動を終了する(S1005:NO)。

10

【0116】

以上、本発明について、その実施の形態に基づき具体的に説明したが、本発明はこれに限定されるものではなく、その要旨を逸脱しない範囲で種々変更可能である。

【0117】

例えば発明の実施形態を、以下のようにとらえることもできる。

【0118】

20

(1) 複数のタスクを時分割で実行する計算機であって、

他の計算機と通信可能に接続され、

前記タスクを実行するタスク処理部と、

前記各タスクについて、前記タスクの状態を示す情報であるタスク制御ブロックを記憶する管理テーブルと、

前記他の計算機から前記管理テーブルを取得する管理テーブル取得部と、

前記タスク処理部が実行しているタスクである実行中タスクを、他のタスクである切替タスクに切り替えるべきタイミングである切替可能ポイントを検出する切替可能ポイント検出部と、

前記切替可能ポイントが検出された場合に、当該計算機が備える前記管理テーブルに記憶されている前記タスク制御ブロックと、前記他の計算機から取得した前記管理テーブルに含まれる前記タスク制御ブロックとに基づいて前記切替タスクを決定し、前記タスク処理部が実行するタスクを前記切替タスクに切り替えるディスパッチャと、を備える計算機。

30

【0119】

(2) (1)に記載の計算機であって、

前記複数の計算機のそれぞれにおいて、前記切替可能ポイントが検出されたか否かを示す切替情報を交換する切替情報交換部を備え、

前記ディスパッチャは、前記切替情報に基づいて前記各計算機が前記切替可能ポイントを検出したか否かを判定し、所定数の前記計算機が前記切替可能ポイントを検出している場合に、前記切替タスクを決定する計算機。(3) (1)に記載の計算機であって、

40

前記切替可能ポイント検出部は、前記実行中タスクが所定の命令を呼び出したか否かにより、前記切替可能ポイントを検出する計算機。

【0120】

(4) (3)に記載の計算機であって、

前記所定の命令は、前記実行中タスク以外のタスクから値を取得する命令であること、を特徴とする計算機。

【0121】

(5) (3)に記載の計算機であって、

前記所定の命令は、ユーザプログラムからオペレーティングシステムに対するシステム

50

コールであること、を特徴とする計算機。

【0122】

(6) (3)に記載の計算機であって、

前記実行中タスクに含まれる条件分岐、前記条件分岐において用いられる変数に値を設定する命令を特定する条件分岐特定部を備え、

前記所定の命令は、前記条件分岐において用いられる変数に値を設定する命令であること、を特徴とする計算機。

【0123】

(7) (1)に記載の計算機であって、

前記管理テーブルには、前記実行中タスクに係る前記タスク制御ブロックを格納する第1の格納領域と、実行可能な状態にある前記タスクに係る前記タスク制御ブロックを格納する第2の格納領域と、待ち状態にある前記タスクに係る前記タスク制御ブロックを格納する第3の格納領域とが含まれ、

前記ディスパッチャは、当該計算機が備える前記管理テーブルに含まれる前記第2の格納領域と、前記他の計算機から取得した前記管理テーブルに含まれる前記第2の格納領域との両方に格納されている前記タスク制御ブロックに対応する前記タスクを前記切替タスクとして決定すること、を特徴とする計算機。

【0124】

(8) (7)に記載の計算機であって、

前記タスク制御ブロックには前記タスクの優先度が設定され、

前記ディスパッチャは、当該計算機が備える前記管理テーブルに含まれる前記第2の格納領域と、前記他の計算機から取得した前記管理テーブルに含まれる前記第2の格納領域との両方に格納されている前記タスク制御ブロックが複数存在する場合、前記タスク制御ブロックの前記優先度に応じて前記切替タスクを決定すること、を特徴とする計算機。

【0125】

(9) (3)に記載の計算機であって、

前記タスクを特定する情報に対応付けて、前記切替可能ポイントを検出するための前記命令を記憶する命令記憶部を備え、

前記切替可能ポイント検出部は、前記実行中タスクに対応する前記命令を前記命令記憶部から読み出し、前記読み出した命令を前記実行中タスクが呼び出すか否かにより、前記切替可能ポイントを検出すること、を特徴とする計算機。

【0126】

(10) (3)に記載の計算機であって、

前記所定の命令は、所定のAPIとして定義されている関数の呼び出しであること、を特徴とする計算機。

【0127】

(11) (1)に記載の計算機であって、

オペレーティングシステムが動作し、

前記オペレーティングシステムはユーザプログラムを実行し、

前記切替可能ポイント検出部は、前記実行中タスクが前記ユーザプログラムに係るタスクである場合に、前記実行中タスクに応じて、前記実行中タスクに含まれている所定の命令の実行前の時点、または、前記所定の命令の実行後の時点のいずれかを前記切替可能ポイントとして決定すること、を特徴とする計算機。

【0128】

(12) (1)に記載の計算機であって、

タスク間通信を行うタスクを特定する情報に対応付けて、前記タスク間通信の通信状態を示す情報を記憶するシグナル管理テーブルと、

前記他の計算機から前記シグナル管理テーブルを取得するシグナル管理テーブル取得部と、を備え、

前記ディスパッチャは、前記切替タスクを決定する際に、当該計算機が備える前記シグ

10

20

30

40

50

ナル管理テーブルに記憶されている前記通信状態を示す情報と、前記他の計算機から取得した前記シグナル管理テーブルに含まれる前記通信状態を示す情報とが一致するタスクを決定すること、を特徴とする計算機。

【0129】

(13) (1)に記載の計算機であって、

第1および第2の前記タスク処理部を有し、

前記ディスパッチャは、

前記第1のタスク処理部について前記切替タスクを決定する場合には、当該計算機が備える前記管理テーブルに記憶されている前記タスク制御ブロックと、前記他の計算機から取得した前記管理テーブルに記憶されている前記タスク制御ブロックとに基づいて、前記切替タスクを決定し、

10

前記第2のタスク処理部について前記切替タスクを決定する場合には、当該計算機が備える前記管理テーブルに記憶されている前記タスク制御ブロックのみに基づいて、前記切替タスクを決定すること、を特徴とする計算機。

【0130】

(14) (13)に記載の計算機であって、

前記タスクの処理に応じて、前記タスクを前記第1および第2のタスク処理部のいずれかに割り当てるタスク割当部を備えること、を特徴とする計算機。

【0131】

(15) (14)に記載の計算機であって、

20

前記第1のタスク処理部に割り当てられている第1のタスクが、当該タスクの割り当てを前記第1のタスク処理部から前記第2のタスク処理部に変更するタイミングを指定する命令である切替可能シグナルを呼び出した場合に、前記ディスパッチャは、前記第1のタスクを他の第2のタスクに切り替え、前記タスク割当部は、前記第1のタスクを前記第2のタスク処理部に割り当てること、を特徴とする計算機。

【0132】

(16) (2)に記載の計算機であって、

前記ディスパッチャは、所定時間内に前記所定数の前記計算機が前記切替可能ポイントを検出したか否かを判定し、

前記所定時間内に前記切替可能ポイントを検出していない他の前記計算機に対して、同期が外れた旨を通知する同期外通知部を備えること、を特徴とする計算機。

30

【0133】

(17) (16)に記載の計算機であって、

前記他の計算機と通信可能に接続される多数決機に通信可能に接続され、

前記同期外通知部はさらに、前記所定時間内に前記切替可能ポイントを検出していない前記他の計算機である同期外計算機を指定して、前記同期外計算機の同期が外れた旨を示す同期外通知を前記多数決機に送信し、

前記多数決機は、

前記同期外通知を受信する同期外通知受信部と、

前記同期外通知に指定された前記同期外計算機以外の前記計算機による演算結果を比較して、同一になった数の最も多い前記演算結果を出力する多数決出力部と、を特徴とする計算機。

40

【0134】

(18) (1)に記載の計算機であって、

前記他の計算機と通信可能に接続される多数決機と通信可能に接続され、

前記多数決機は、

前記計算機による演算結果を比較して、同一になった数の最も多い前記演算結果である多数決結果を出力する多数決出力部と、

前記多数決結果と異なる前記演算結果を出力した前記計算機を前記各計算機に通知するエラー通知部とを備え、

50

前記ディスパッチャは、当該計算機が備える前記管理テーブルに記憶されている前記タスク制御ブロックと、前記多数決結果と異なる前記演算結果を出力した前記計算機以外の前記計算機から取得した前記管理テーブルに記憶されている前記タスク制御ブロックとに基づいて、前記切替タスクを決定すること、を特徴とする計算機。

【0135】

(19) (1)に記載の計算機であって、

計算機個別のエラーが発生したとき、エラー情報を計算機間で通信し、計算機間で同期をとってエラー処理を行うこと、を特徴とする計算機。

【0136】

(20) (1)に記載の計算機であって、

タスクにおいてプログラムエラーが発生したとき、ディスパッチャが当該エラーを同期して処理すること、を特徴とする計算機。

【0137】

(21)複数のタスクを時分割で実行する計算機の制御方法であって、

他の前記計算機と通信可能に接続され、前記タスクを実行するタスク処理部と、前記各タスクについて、前記タスクの状態を示す情報であるタスク制御ブロックを記憶する管理テーブルとを備える計算機が、

前記タスク処理部が実行しているタスクである実行中タスクを、他のタスクである切替タスクに切り替えるべきタイミングである切替可能ポイントを検出し、

前記他の計算機から前記管理テーブルを取得し、

前記切替可能ポイントが検出された場合に、当該計算機が備える前記管理テーブルに記憶されている前記タスク制御ブロックと、前記他の計算機から取得した前記管理テーブルに含まれる前記タスク制御ブロックとに基づいて前記切替タスクを決定し、

前記タスク処理部が実行するタスクを前記切替タスクに切り替えること、を特徴とする計算機の制御方法。

【0138】

(22)複数のタスクを時分割で実行するためのプログラムであって、

他の前記計算機と通信可能に接続され、前記タスクを実行するタスク処理部と、前記各タスクについて、前記タスクの状態を示す情報であるタスク制御ブロックを記憶する管理テーブルとを備える計算機に、

前記タスク処理部が実行しているタスクである実行中タスクを、他のタスクである切替タスクに切り替えるべきタイミングである切替可能ポイントを検出するステップと、

前記他の計算機から前記管理テーブルを取得するステップと、

前記切替可能ポイントが検出された場合に、当該計算機が備える前記管理テーブルに記憶されている前記タスク制御ブロックと、前記他の計算機から取得した前記管理テーブルに含まれる前記タスク制御ブロックとに基づいて、前記切替タスクを決定するステップと、

前記タスク処理部が実行するタスクを前記切替タスクに切り替えるステップと、を実行させるためのプログラム。

【0139】

(23) (15)に記載の計算機であって、

前記タスクの状態を示す情報であるタスク制御ブロックを記憶する前記管理テーブルには、前記実行中タスクに係る前記タスク制御ブロックを格納する第1の格納領域と、実行可能な状態にある前記タスクに係る前記タスク制御ブロックを格納する第2の格納領域と、待ち状態にある前記タスクに係る前記タスク制御ブロックを格納する第3の格納領域と、前記第1のタスク処理部に割り当てられている第1のタスクが、当該タスクの割り当てを前記第1のタスク処理部から前記第2のタスク処理部に変更するタイミングの順序を格納する第4の格納領域と、を備える計算機。

【0140】

(24) (23)に記載の計算機であって、

前記第1のタスク処理部に割り当てられている第1のタスクが、当該タスクの割り当てを前記第1のタスク処理部から前記第2のタスク処理部に変更するタイミングを指定する命令である切替可能シグナルを呼び出した場合に、前記ディスパッチャは、前記第1のタスクを他の第2のタスクに切り替え、前記第1のタスク処理部に割り当てられている第1のタスクが、当該タスクの割り当てを前記第1のタスク処理部から前記第2のタスク処理部に変更するタイミングの順序を格納する第4の格納領域に追加登録し、前記タスク割当部は、前記第4の格納領域にて実行順序が訪れたときのみ前記第1のタスクを前記第2のタスク処理部に割り当てること、

10

【0141】

(25) (23)に記載の計算機であって、

前記第2のタスク処理部に割り当てられている第1のタスクが、当該タスクの割り当てを前記第2のタスク処理部から前記第1のタスク処理部に変更するタイミングを指定する命令である切替可能シグナルを呼び出した場合に、前記ディスパッチャは、前記第4の格納領域から当該タスクの前記順序情報を削除し、前記第2のタスク処理部に割り当てられている第1のタスクが、当該タスクの割り当てを前記第2のタスク処理部から前記第1のタスク処理部に割り当てること、

を特徴とする計算機。

20

【0142】

(26) (15)に記載の計算機であって、

前記第1のタスク処理部と第2のタスク処理部で実行するメモリ回収タスクと、前記メモリ回収タスクが起動するメモリ空き容量状況の閾値を格納するタスク処理部毎メモリ回収閾値管理格納部と、

を有する特徴とする計算機。

【0143】

(27) (26)に記載の計算機であって、

前記タスク処理部毎メモリ回収閾値管理処理部に、前記同期タスクが実行される処理部のメモリ回収閾値よりも他のタスク処理部でのメモリ回収閾値が先に閾値条件に合致する閾値有し、同期タスク実行処理部より先に他のタスク処理部にてメモリ回収を実行すること、

30

を特徴とする計算機。

【符号の説明】

【0144】

10 計算機システム

100A、100B 計算機

110A、110B メモリ

120A、120B タスクプログラム

140A、140B オペレーティングシステム

141A、141B ディスパッチャ

40

142A、142B タスク間メッセージ管理テーブル

143A-1、143A-2 タスク管理テーブル

143B-1、143B-2 タスク管理テーブル

111A、111B 同期計算機管理テーブル

112A、112B シグナル管理テーブル

144 実行可能タスクキュー

145 待機状態タスクキュー

146 TCB

147 実行中タスク

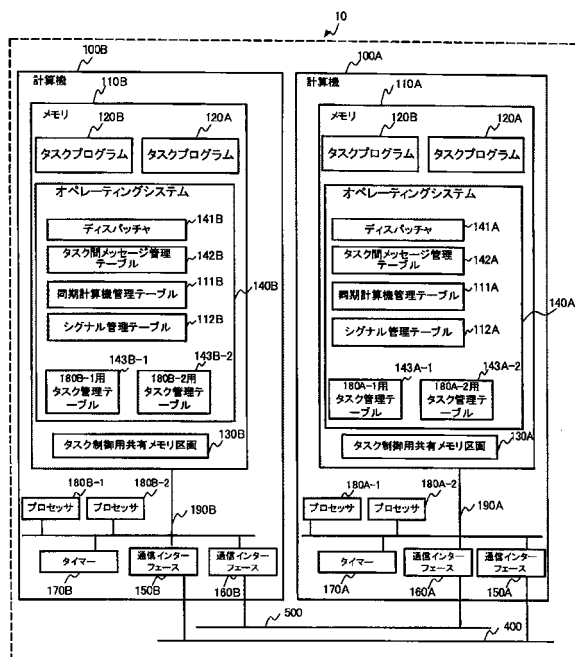
130A、130B タスク制御用共有メモリ区画

50

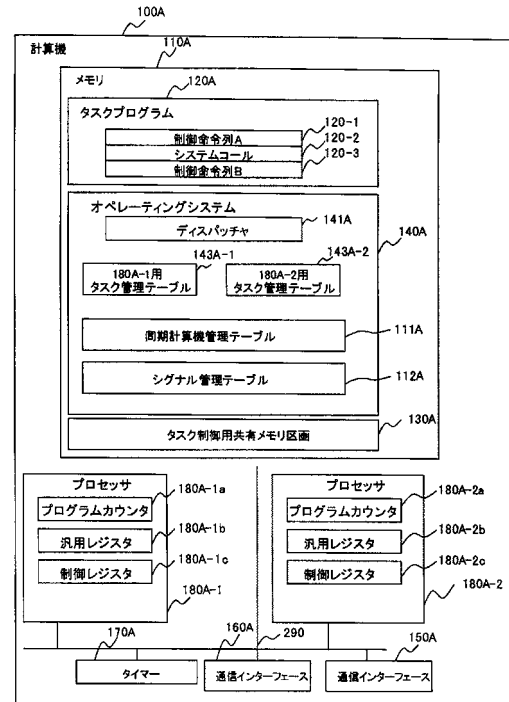
- 150A、150B 通信インターフェース
 160A、160B 通信インターフェース
 170A、170B タイマー
 180A-1、180A-2、180B-1、180B-2 プロセッサ
 180A-1a、180A-2a プログラムカウンタ
 180A-1b、180A-2b 汎用レジスタ
 180A-1c、180A-2c 制御レジスタ
 190A、190B バス
 148 同期タスク順序管理キュー
 149 同期タスク順序管理用ポインタ
 120C メモリ回収タスクプログラム
 113B プロセッサ毎メモリ回収閾値管理テーブル
 600 記録媒体
 610 同期タスク非同期タスク定義テーブル

10

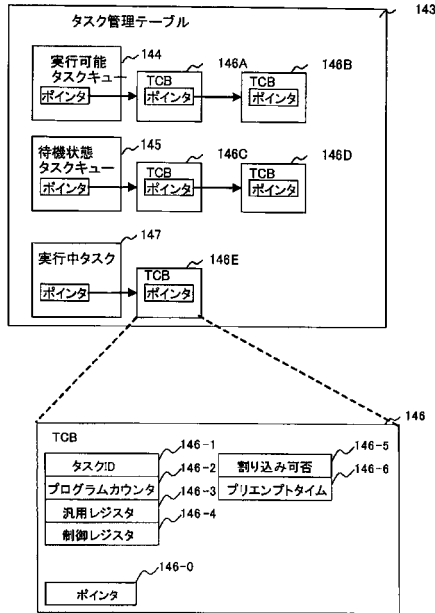
【図 1】



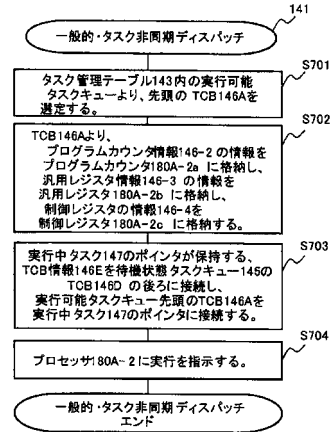
【図 2】



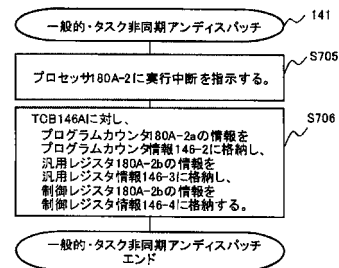
【図 3】



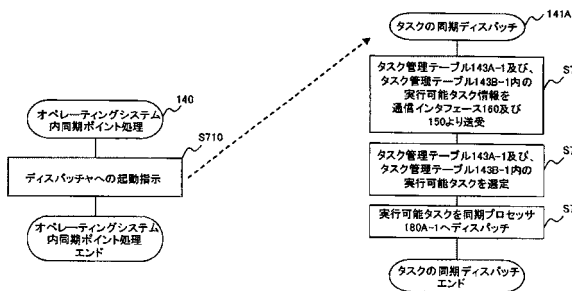
【図 4】



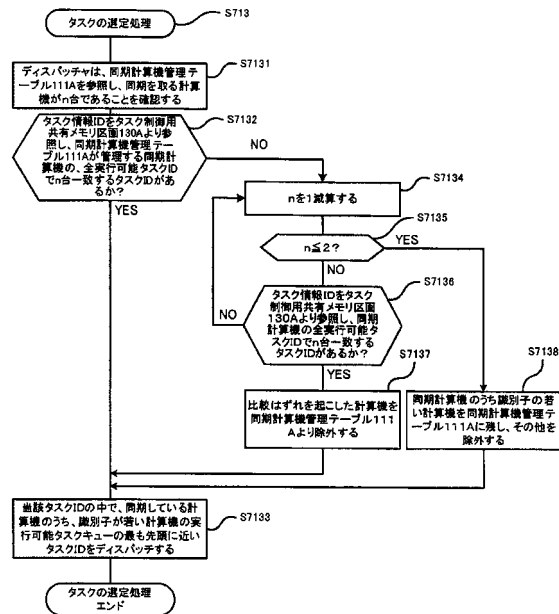
【図 5】



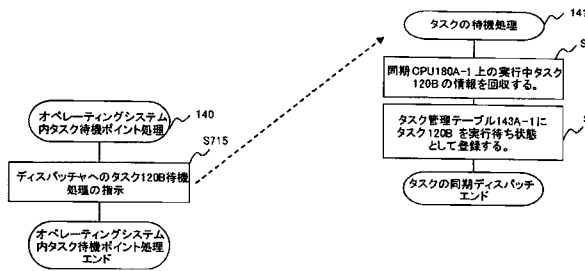
【図 6】



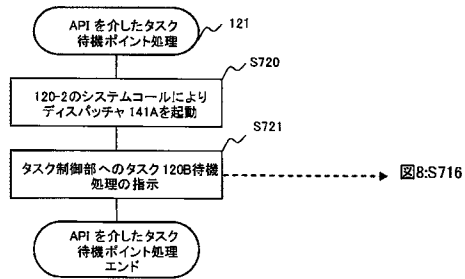
【図 7】



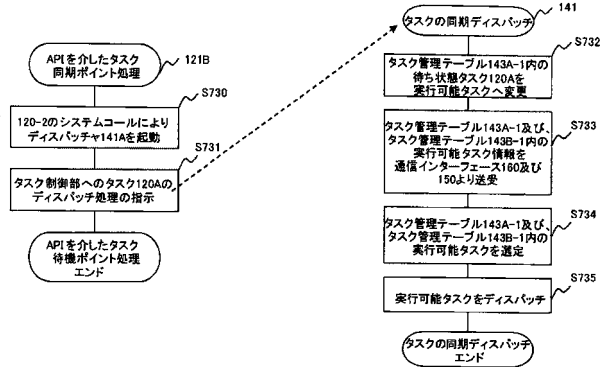
【図 8】



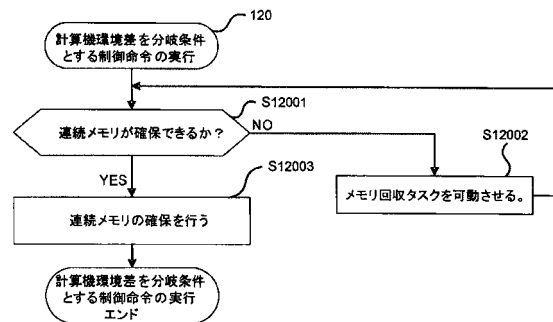
【図 9】



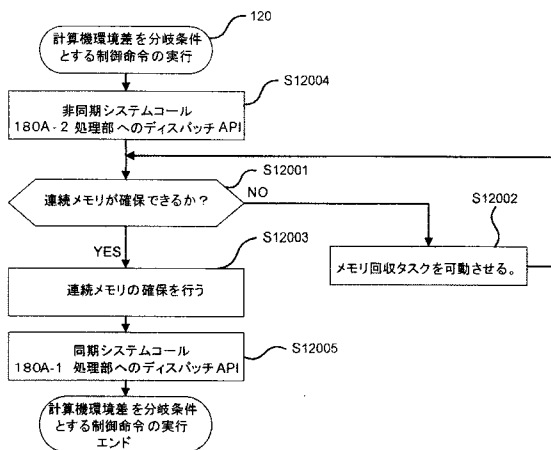
【図 10】



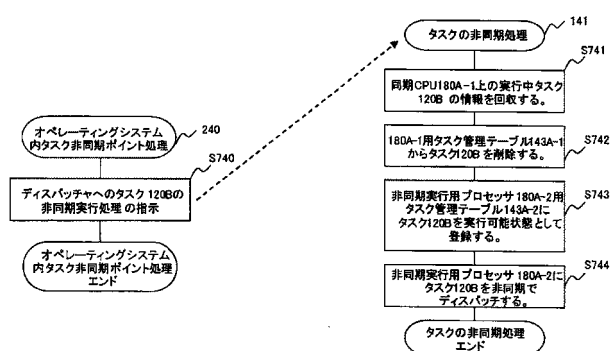
【図 11】



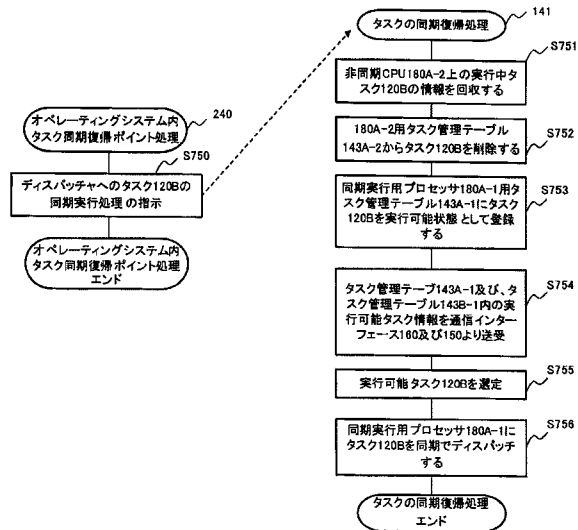
【図 12】



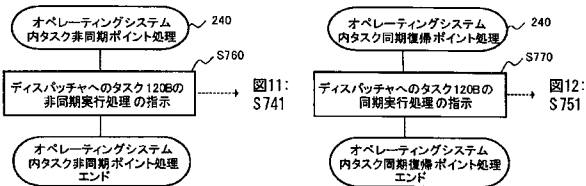
【図 13】



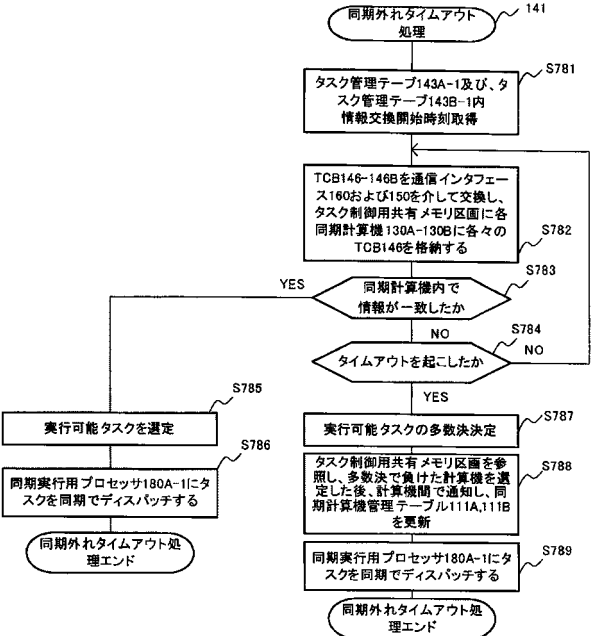
【図14】



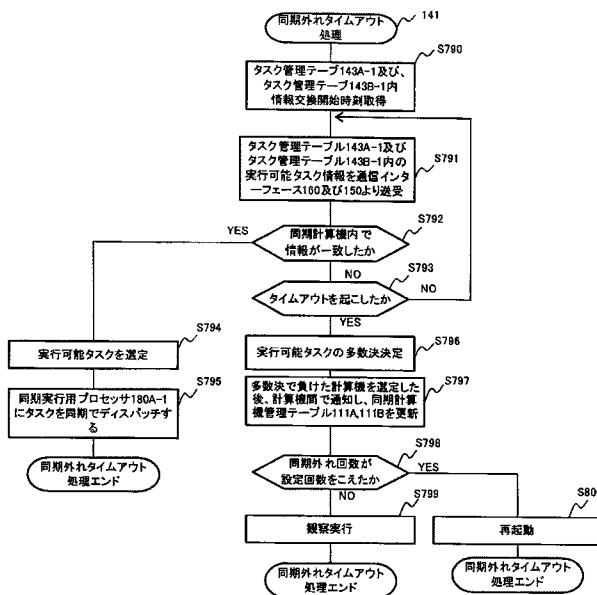
【図15】



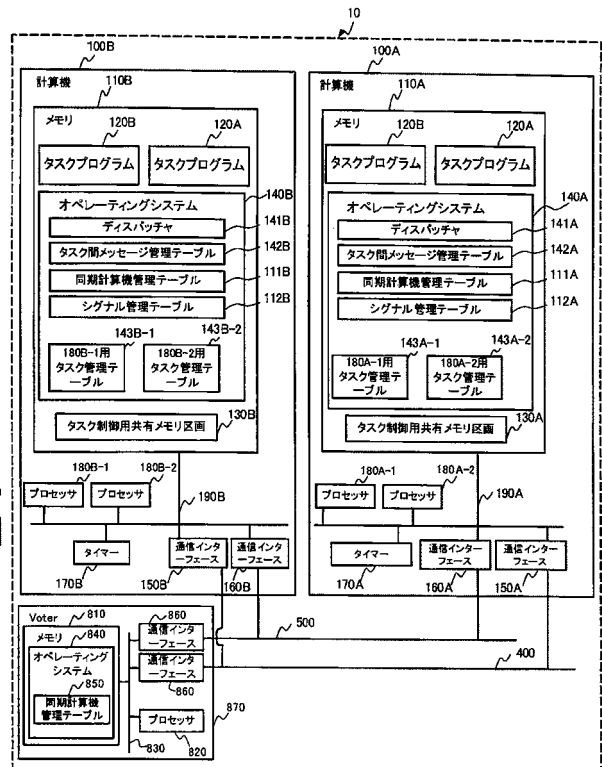
【図16】



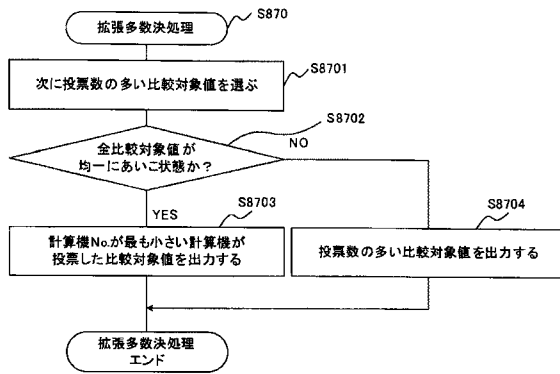
【図17】



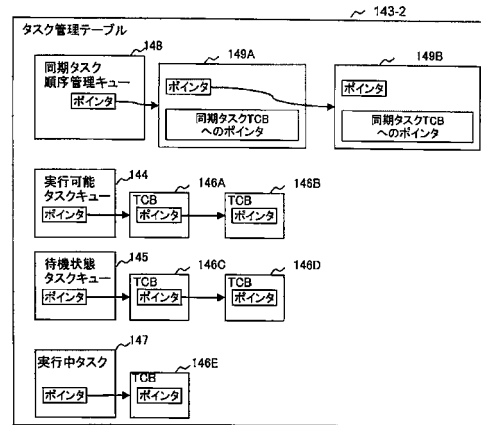
【図18】



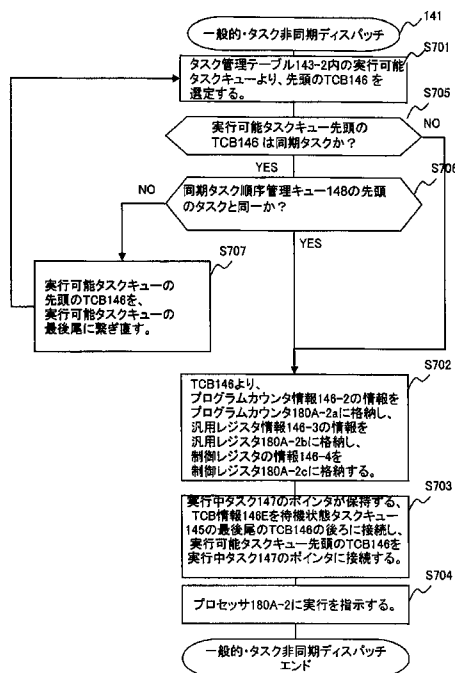
【図 24】



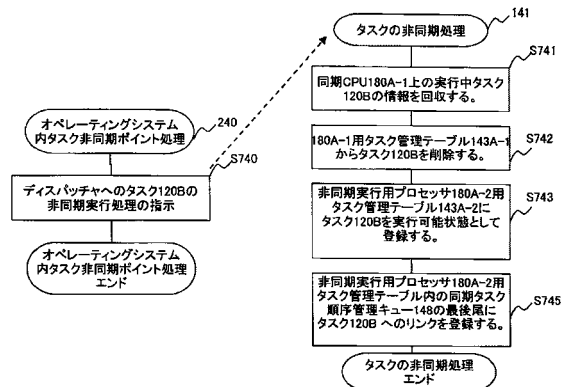
【図 25】



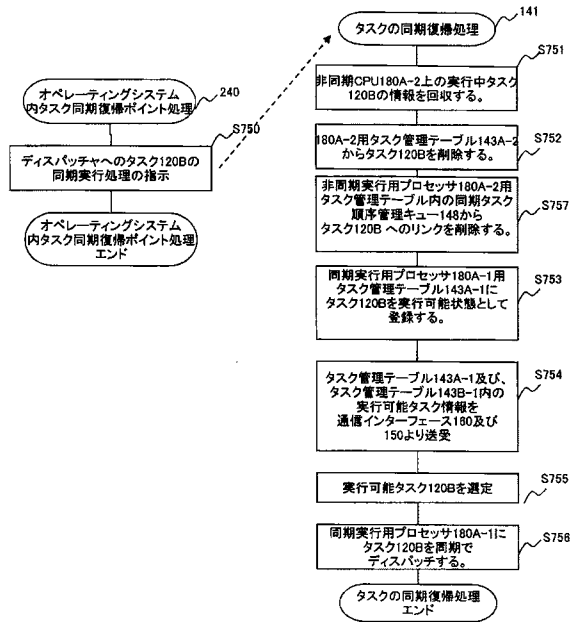
【図 26】



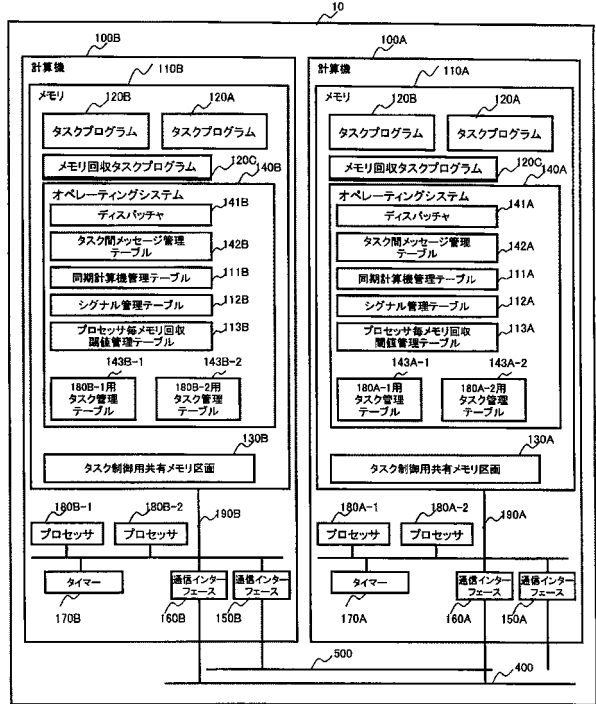
【図 27】



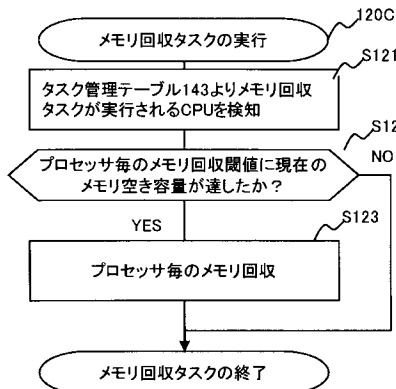
【図 28】



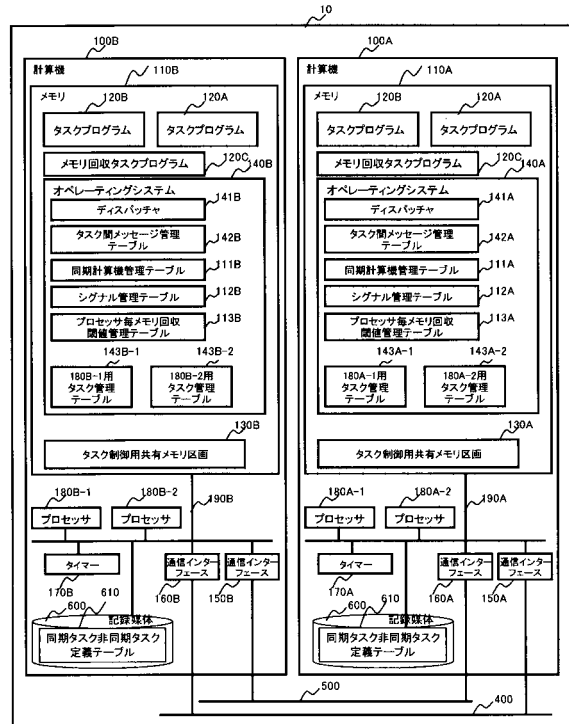
【図 29】



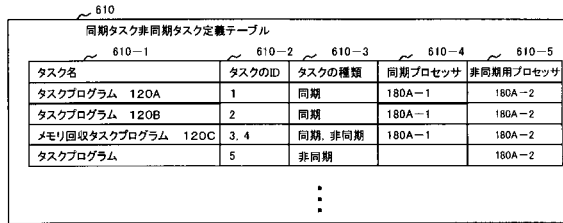
【図 30】



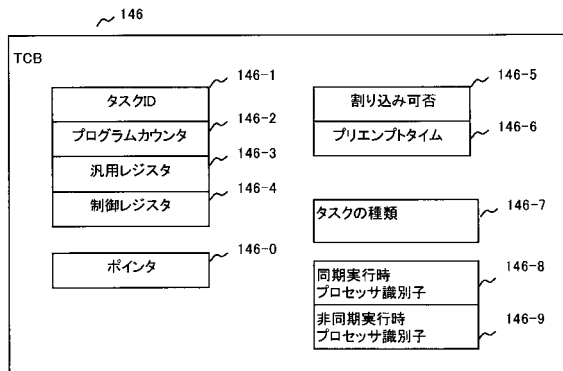
【図 31】



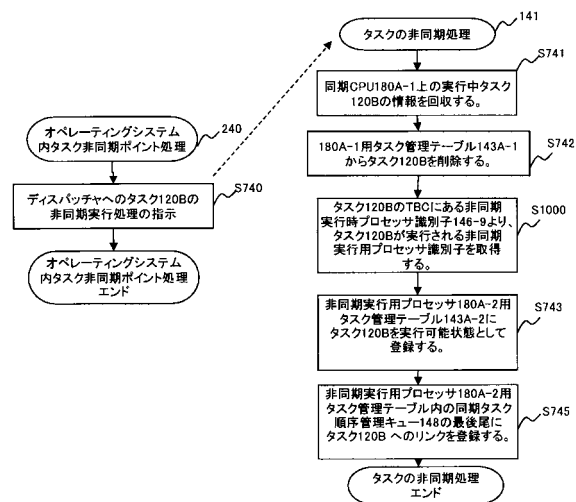
【図 3 2】



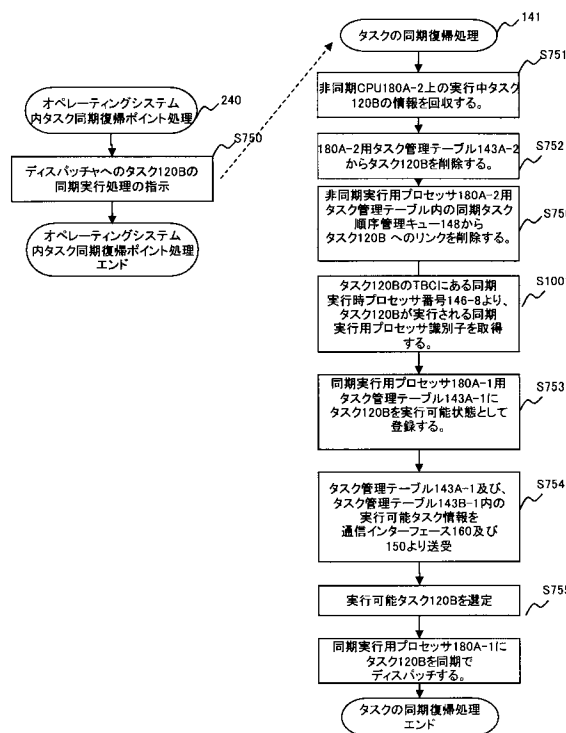
【図 3 3】



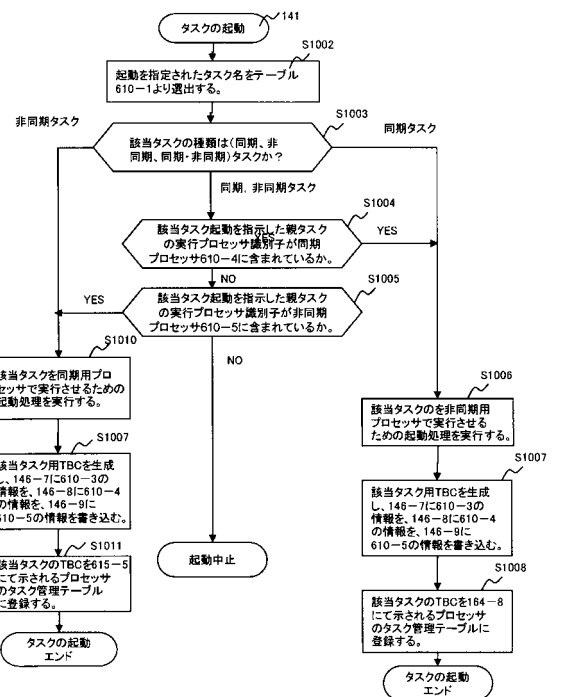
【図 3 4】



【図 3 5】



【図 3 6】



フロントページの続き

(72)発明者 大平 崇博

茨城県日立市大みか町五丁目2番1号 株式会社日立製作所 情報制御システム社内

(72)発明者 武田 直之

茨城県日立市大みか町五丁目2番1号 株式会社日立製作所 情報制御システム社内

審査官 漆原 孝治

(56)参考文献 特開2009-003537 (JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 9/46 - 9/54