

(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2008-242510

(P2008-242510A)

(43) 公開日 平成20年10月9日(2008.10.9)

(51) Int.Cl.

G 0 6 F 9/48 (2006.01)

F I

G O 6 F 9/46 4 5 2 E

テーマコード (参考)

審査請求 未請求 請求項の数 17 O L (全 16 頁)

(21) 出願番号 特願2007-77633 (P2007-77633)
 (22) 出願日 平成19年3月23日 (2007. 3. 23)

(71) 出願人 000004237
 日本電気株式会社
 東京都港区芝五丁目7番1号
 (74) 代理人 100109313
 弁理士 机 昌彦
 (74) 代理人 100121290
 弁理士 木村 明隆
 (74) 代理人 100111637
 弁理士 谷澤 靖久
 (72) 発明者 伊藤 幹雄
 東京都港区芝五丁目7番1号
 日本電気株式会社内

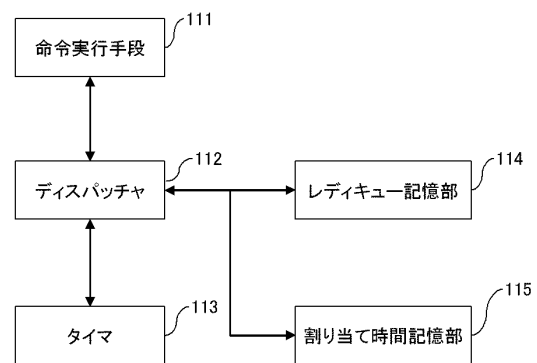
(54) 【発明の名称】 ジョブへの資源割当方式、ジョブへの資源割当方法およびプログラム

(57) 【要約】

【課題】 単一システムで複数のジョブを実行する場合において、特定のジョブの負荷が増大した場合に、他のジョブが実行出来ない状況や、他のジョブへのシステムの割り当て時間が減少する状況が発生するという課題があった。

【解決手段】 あらかじめ定められたジョブ毎のプロセス優先割り当て時間に基づいて、優先期間にある優先ジョブの切替を通知するタイマ113と、通知に基づいて、優先ジョブとして選択したジョブに対応する、情報処理システムが実行可能なプロセスをジョブ毎に待ち行列にしたレディーキューから、レディーキューの先頭のプロセスを取り出して、命令実行手段111に割り当てるディスパッチャ112と、割り当てられたプロセスを実行中プロセスとして実行中プロセスの命令を実行する命令実行手段111とを有する。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

情報処理システムにおいて、予め定められたジョブ毎のプロセッサ優先割り当て時間に基づいてジョブの切替を通知するタイマと、プロセスの命令を実行する命令実行手段と、前記タイマからの通知に基づいて優先ジョブとして選択したジョブに対応するプロセスを含むレディーキューから、先頭のプロセスを取り出して前記命令実行手段に割り当てて当該プロセスの命令を実行させるディスパッチャと、を備えたことを特徴とするジョブへの資源割当方式。

【請求項 2】

前記ディスパッチャは、前記優先ジョブとして選択したジョブの前記レディーキューが空であった場合は、次の優先ジョブを選択することを特徴とする請求項 1 記載のジョブへの資源割当方式。

10

【請求項 3】

前記プロセッサ優先割り当て時間は、一定時間の周期を前記ジョブ毎のシステム割り当て割合で分配した時間であることを特徴とする請求項 1 または 2 記載のジョブへの資源割当方式。

【請求項 4】

前記ディスパッチャは、所定の処理を実行する例外プロセスを含む例外レディーキューに前記例外プロセスが存在する場合には、前記優先ジョブよりも前記例外プロセスに対応する最優先ジョブを優先して、前記例外レディーキューの先頭の例外プロセスを取り出して前記命令実行手段に割り当てて当該例外プロセスの命令を実行させることを特徴とする請求項 1 乃至 3 のいずれかに記載のジョブへの資源割当方式。

20

【請求項 5】

前記ディスパッチャは、前記優先ジョブに対応するプロセス間で優先順位が変化した場合は、実行中のプロセスに対応する前記レディーキューに戻し、当該レディーキューの先頭のプロセスを取り出して前記命令実行手段に割り当てて当該プロセスの命令を実行させることを特徴とする請求項 1 乃至 4 のいずれかに記載のジョブへの資源割当方式。

【請求項 6】

前記ディスパッチャは、前記最優先ジョブに対応する例外プロセス間で優先順位が変化した場合は、実行中の例外プロセスに対応する前記例外レディーキューに戻し、当該例外レディーキューの先頭の例外プロセスを取り出して前記命令実行手段に割り当てて当該例外プロセスの命令を実行させることを特徴とする請求項 4 または 5 に記載のジョブへの資源割当方式。

30

【請求項 7】

前記タイマは、前記命令実行手段がプロセスの命令を一つ実行する度に、実行中のプロセスに対応するジョブの優先期間の計時値を更新し、当該ジョブの前記プロセッサ優先割り当て時間に達したか否かを確認することを特徴とする請求項 1 乃至 6 のいずれかに記載のジョブへの資源割当方式。

【請求項 8】

前記ジョブの終了、追加の発生に対応して前記プロセッサ優先割り当て時間を再算出する手段を備えたことを特徴とする請求項 1 乃至 7 のいずれかに記載のジョブへの資源割当方式。

40

【請求項 9】

情報処理システムにおけるジョブへの資源割り当て方法であって、予め定められたジョブ毎のプロセッサ優先割り当て時間に基づいてジョブの切替を通知し、前記通知に基づいて優先ジョブとして選択したジョブに対応するプロセスを含むレディーキューから、先頭のプロセスを取り出して当該プロセスの命令を実行することを特徴とするジョブへの資源割当方法。

【請求項 10】

前記通知に基づいて優先ジョブとして選択したジョブの前記レディーキューが空であった

50

場合は、次の優先ジョブを選択することを特徴とする請求項 9 記載のジョブへの資源割当方法。

【請求項 11】

前記プロセッサ優先割り当て時間は、一定時間の周期を前記ジョブ毎のシステム割り当て割合で分配した時間であることを特徴とする請求項 9 または 10 記載のジョブへの資源割当方法。

【請求項 12】

所定の処理を実行する例外プロセスを含む例外レディーキューに前記例外プロセスが存在する場合には、前記優先ジョブよりも前記例外プロセスに対応する最優先ジョブを優先して、前記例外レディーキューの先頭の例外プロセスを取り出して当該例外プロセスの命令を実行することを特徴とする請求項 9 乃至 11 のいずれかに記載のジョブへの資源割当方法。

10

【請求項 13】

前記優先ジョブに対応するプロセス間で優先順位が変化した場合は、実行中のプロセスに対応する前記レディーキューに戻し、当該レディーキューの先頭のプロセスを取り出して当該プロセスの命令を実行することを特徴とする請求項 9 乃至 12 のいずれかに記載のジョブへの資源割当方法。

【請求項 14】

前記最優先ジョブに対応する例外プロセス間で優先順位が変化した場合は、実行中の例外プロセスに対応する前記例外レディーキューに戻し、当該例外レディーキューの先頭の例外プロセスを取り出して当該例外プロセスの命令を実行することを特徴とする請求項 12 または 13 のいずれかに記載のジョブへの資源割当方法。

20

【請求項 15】

前記プロセスの命令を一つ実行する度に、実行中のプロセスに対応するジョブの優先期間の計時値を更新し、当該ジョブの前記プロセッサ優先割り当て時間に達したか否かを確認することを特徴とする請求項 9 乃至 14 のいずれかに記載のジョブへの資源割当方法。

【請求項 16】

前記ジョブの終了、追加の発生に対応して前記プロセッサ優先割り当て時間を再算出する手段を備えたことを特徴とする請求項 9 乃至 15 のいずれかに記載のジョブへの資源割当方法。

30

【請求項 17】

情報処理システムに、予め定められたジョブ毎のプロセッサ優先割り当て時間に基づいてジョブの切替を通知し、前記通知に基づいて優先ジョブとして選択したジョブに対応するプロセスを含むレディーキューから、先頭のプロセスを取り出して当該プロセスの命令を実行する処理を行わせることを特徴とするプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ジョブへの資源割当方式に関し、特に複数のジョブを単一システムで実行する場合に負荷のバランスを調整するジョブへの資源割当方式、ジョブへの資源割当方法およびプログラムに関する。

40

【背景技術】

【0002】

従来のジョブへの資源割当の方法として、当該ジョブに対する CPU の割り当て時間を保証するスケジュール方法を提供するものがある（例えば、特許文献 1）。この従来のスケジュール方法は、CPU 割り当て保証対象ジョブを予め定め、当該ジョブの CPU 使用時間を計測し、予め定めた単位時間内に一定時間以上の CPU が割り当てられていなかった場合、優先的に CPU を割り当てるものである。これを実現するために、ジョブ開始時に当該ジョブが CPU 割り当て保証対象ジョブであるか否かを判断し、CPU 割り当て保証対象ジョブであると判断された場合は、当該ジョブのジョブ管理テーブルを CPU 割り当

50

て保証ジョブ管理テーブルに登録し、次に、当該ジョブが共用資源待ちになるか否かを判断し、共用資源待ちになると判断された場合には、先に共用資源を確保しているジョブがCPU割り当て保証対象ジョブであるか否かを判断し、CPU割り当て保証対象ジョブでないと判断した場合には一時優先フラグを設定した当該ジョブのジョブ管理テーブルをCPU割り当て保証ジョブ管理テーブルに登録し、共用資源を解放させる処理を行う。また、タイマ処理部から起動されたCPU割り当て保証制御部が、当該ジョブが起動されてからのCPU使用時間、CPU割り当て保証時間、CPU割り当て監視時間を参照して優先スケジュールキューを更新し、その後、スケジューラにて当該ジョブへのCPU割り当て処理を行う構成を有している。

【0003】

10

【特許文献1】特開2000-148512

【発明の開示】

【発明が解決しようとする課題】

【0004】

しかしながら、この従来の方法では、ジョブをスケジューリングしてCPUを割り当てるためには、上述したようにジョブ管理テーブルを作成してCPU割り当て保証ジョブ管理テーブルに登録するというような煩雑な処理を実行する必要がある、その結果としてオーバーヘッドが大きく汎用性に欠けるという問題があった。

【0005】

本発明の目的は、複数のジョブを単一のシステムで実行する場合において、オーバーヘッドの少ないシンプルな構成で汎用性に優れたジョブへの資源割当方式、ジョブへの資源割当方法およびプログラムを提供することにある。

20

【課題を解決するための手段】

【0006】

本発明の第1のジョブへの資源割当方式は、情報処理システムにおいて、予め定められたジョブ毎のプロセッサ優先割り当て時間に基づいてジョブの切替を通知するタイマと、プロセスの命令を実行する命令実行手段と、前記タイマからの通知に基づいて優先ジョブとして選択したジョブに対応するプロセスを含むレディーキューから、先頭のプロセスを取り出して前記命令実行手段に割り当てて当該プロセスの命令を実行させるディスパッチャと、を備えた。

30

【0007】

本発明の第2のジョブへの資源割当方式は、前記第1のジョブへの資源割当方式に於いて、前記ディスパッチャは、前記優先ジョブとして選択したジョブの前記レディーキューが空であった場合は、次の優先ジョブを選択する。

【0008】

本発明の第3のジョブへの資源割当方式は、前記第1または第2のジョブへの資源割当方式に於いて、前記プロセッサ優先割り当て時間は、一定時間の周期を前記ジョブ毎のシステム割り当て割合で分配した時間である。

【0009】

本発明の第4のジョブへの資源割当方式は、前記第1乃至第3のいずれかのジョブへの資源割当方式に於いて、前記ディスパッチャは、所定の処理を実行する例外プロセスを含む例外レディーキューに前記例外プロセスが存在する場合には、前記優先ジョブよりも前記例外プロセスに対応する最優先ジョブを優先して、前記例外レディーキューの先頭の例外プロセスを取り出して前記命令実行手段に割り当てて当該例外プロセスの命令を実行させる。

40

【0010】

本発明の第5のジョブへの資源割当方式は、前記第1乃至第4のいずれかのジョブへの資源割当方式に於いて、前記ディスパッチャは、前記優先ジョブに対応するプロセス間で優先順位が変化した場合は、実行中のプロセスに対応する前記レディーキューに戻し、当該レディーキューの先頭のプロセスを取り出して前記命令実行手段に割り当てて当該プロ

50

セスの命令を実行させる。

【0011】

本発明の第6のジョブへの資源割当方式は、前記第4乃至第5のジョブへの資源割当方式に於いて、前記ディスパッチャは、前記最優先ジョブに対応する例外プロセス間で優先順位が変化した場合は、実行中の例外プロセスに対応する前記例外レディーキューに戻し、当該例外レディーキューの先頭の例外プロセスを取り出して前記命令実行手段に割り当てて当該例外プロセスの命令を実行させる。

【0012】

本発明の第7のジョブへの資源割当方式は、前記第1乃至第6のいずれかのジョブへの資源割当方式に於いて、前記タイマは、前記命令実行手段がプロセスの命令を一つ実行する度に、実行中のプロセスに対応するジョブの優先期間の計時値を更新し、当該ジョブの前記プロセッサ優先割り当て時間に達したか否かを確認する。

10

【0013】

本発明の第8のジョブへの資源割当方式は、前記第1乃至第7のいずれかのジョブへの資源割当方式に於いて、前記ジョブの終了、追加の発生に対応して前記プロセッサ優先割り当て時間を再算出する手段を備えた。

【0014】

本発明の第1のジョブへの資源割当方法は、情報処理システムにおけるジョブへの資源割り当て方法であって、予め定められたジョブ毎のプロセッサ優先割り当て時間に基づいてジョブの切替を通知し、前記通知に基づいて優先ジョブとして選択したジョブに対応するプロセスを含むレディーキューから、先頭のプロセスを取り出して当該プロセスの命令を実行する。

20

【0015】

本発明の第2のジョブへの資源割当方法は、前記第1のジョブへの資源割当方法に於いて、前記通知に基づいて優先ジョブとして選択したジョブの前記レディーキューが空であった場合は、次の優先ジョブを選択する。

【0016】

本発明の第3のジョブへの資源割当方法は、前記第1または第2のジョブへの資源割当方法に於いて、前記プロセッサ優先割り当て時間は、一定時間の周期を前記ジョブ毎のシステム割り当て割合で分配した時間である。

30

【0017】

本発明の第4のジョブへの資源割当方法は、前記第1乃至第3のいずれかのジョブへの資源割当方法に於いて、所定の処理を実行する例外プロセスを含む例外レディーキューに前記例外プロセスが存在する場合には、前記優先ジョブよりも前記例外プロセスに対応する最優先ジョブを優先して、前記例外レディーキューの先頭の例外プロセスを取り出して当該例外プロセスの命令を実行する。

【0018】

本発明の第5のジョブへの資源割当方法は、前記第1乃至第4のいずれかのジョブへの資源割当方法に於いて、前記優先ジョブに対応するプロセス間で優先順位が変化した場合は、実行中のプロセスに対応する前記レディーキューに戻し、当該レディーキューの先頭のプロセスを取り出して当該プロセスの命令を実行する。

40

【0019】

本発明の第6のジョブへの資源割当方法は、前記第1乃至第5のいずれかのジョブへの資源割当方法に於いて、前記最優先ジョブに対応する例外プロセス間で優先順位が変化した場合は、実行中の例外プロセスに対応する前記例外レディーキューに戻し、当該例外レディーキューの先頭の例外プロセスを取り出して当該例外プロセスの命令を実行する。

【0020】

本発明の第7のジョブへの資源割当方法は、前記第1乃至第6のいずれかのジョブへの資源割当方法に於いて、前記プロセスの命令を一つ実行する度に、実行中のプロセスに対応するジョブの優先期間の計時値を更新し、当該ジョブの前記プロセッサ優先割り当て時

50

間に達したか否かを確認する。

【 0 0 2 1 】

本発明の第 8 のジョブへの資源割当方法は、前記第 1 乃至第 7 のいずれかのジョブへの資源割当方法に於いて、前記ジョブの終了、追加の発生に対応して前記プロセッサ優先割り当て時間を再算出する手段を備えた。

【 0 0 2 2 】

本発明のプログラムは、情報処理システムに、予め定められたジョブ毎のプロセッサ優先割り当て時間に基づいてジョブの切替を通知し、前記通知に基づいて優先ジョブとして選択したジョブに対応するプロセスを含むレディーキューから、先頭のプロセスを取り出して当該プロセスの命令を実行する処理を行わせる。

10

【発明の効果】

【 0 0 2 3 】

本発明によれば、複数のジョブを単一のシステムで実行する場合において、特定のジョブの負荷が増大した場合でも、他のジョブの性能低下や実行出来ない状況が発生することを防止する、オーバヘッドの少ないシンプルな構成で汎用性に優れたジョブへの資源割当方式を提供することが可能になる。その理由は、あらかじめ定められたジョブ毎のプロセッサ優先割り当て時間に基づいて、ジョブの優先期間の切替を通知し、当該通知に基づいて、優先期間に実行されるジョブとして選択されたジョブの、プロセッサが実行可能なプロセスを待ち行列にしたレディーキューの先頭からプロセスを取り出して、プロセッサに割り当てて実行させるようにしたからである。

20

【発明を実施するための最良の形態】

【 0 0 2 4 】

次に、本発明について図面を参照して詳細に説明する。

【 0 0 2 5 】

図 1 を参照すると、本発明の第 1 の実施の形態は、命令実行手段 1 1 1 と、ディスパッチャ 1 1 2 と、タイマ 1 1 3 と、レディーキュー記憶部 1 1 4 と、割り当て時間記憶部 1 1 5 とで構成されている。

【 0 0 2 6 】

命令実行手段 1 1 1 は、ディスパッチャ 1 1 2 が選択した実行可能プロセスを構成する命令を実行する。

30

【 0 0 2 7 】

ディスパッチャ 1 1 2 は、図 3 に示す実行可能プロセス群 3 1 1 の中から、命令を実行すべき、当該実行可能プロセスを一つ選択する。ディスパッチャ 1 1 2 は、タイマ 1 1 3 にジョブの優先期間の割り当て時間を送信する。ディスパッチャ 1 1 2 は、タイマ 1 1 3 から切り替え指示を受信すると、当該優先期間のジョブを切り替える。

【 0 0 2 8 】

タイマ 1 1 3 は、当該割り当て時間を受信すると、ジョブの優先期間を計時し、当該優先期間が割り当て時間に達するとディスパッチャ 1 1 2 に切り替え指示を出力する。

【 0 0 2 9 】

レディーキュー記憶部 1 1 4 は、ジョブ毎に実行可能プロセスを待ち行列にしたレディーキュー群を記憶する。

40

【 0 0 3 0 】

図 3 を参照すると、レディーキュー群 3 0 3 は、レディーキュー 3 0 4 ~ 3 0 9 およびレディーキュー 3 1 0 を含む。レディーキュー 3 0 4 ~ 3 0 9 は、それぞれジョブ A ~ F に対応し、実行可能プロセス群 3 1 1 の中で対応するジョブ A ~ F の配下となる実行可能プロセスの待ち行列を構成する。例えば、レディーキュー 3 0 6 は、ジョブ C の配下となる実行可能プロセスの待ち行列を構成する。

【 0 0 3 1 】

レディーキュー 3 1 0 は、最優先ジョブに対応し、配下となる実行可能例外プロセスの待ち行列を構成する。最優先ジョブとは、ジョブ A ~ F に対応していないシステム共通処

50

理、あるいは、ジョブ A ~ F に対応しているが緊急性を要する処理のプロセスを例外プロセスとして定義し、例外プロセスの集合を示す。

【0032】

レディーキュー 310 に実行可能プロセスが繋がれると、ディスパッチャ 112 はその先頭に繋がれた実行可能プロセスを最優先で選択する。命令実行手段 111 は、選択された実行可能プロセスの命令を実行する。最優先ジョブには、ジョブ A ~ F への割り当て割合 202 ~ 207 と対応するものは定義されないため、プロセッサ優先割り当て時間に相当するものはない。

【0033】

実行可能プロセス群 311 は、ディスパッチャ 112 から選択され、命令実行手段 111 により実行されるのを待っているプロセス群である。

10

【0034】

一般的に、プロセスがレディーキューに繋がって待ち行列を構成し、レディーキューから取り出されて実行される処理は、当業者にとって周知の技術であるが、本実施の形態の理解をより容易にする上で有用な点について説明する。本実施の形態においては、命令実行手段 111 による実行状態にあるプロセスを実行中プロセスと呼ぶ。ある時点での実行中プロセスは一つだけである。プロセスは、ジョブ A ~ F、あるいは、最優先ジョブ内での実行順序に関する優先順位を持っている。

【0035】

レディーキュー 304 ~ 309 およびレディーキュー 310 は、各々のレディーキューにおいて、この優先順位に従った、先入れ先出し方式により実行可能プロセスの待ち行列を構成する。優先順位が高いプロセスは低いプロセスより対応するレディーキューの先頭側に繋がれ、同一優先順位のプロセス同士であれば、時間的に先に繋がれたプロセスが対応するレディーキューの先頭側に繋がれる。

20

【0036】

従って、レディーキュー 304 ~ 309 およびレディーキュー 310 の先頭に繋がれた実行可能プロセスは、そのレディーキュー 304 ~ 309 およびレディーキュー 310 を構成する実行可能プロセスの中では、一番優先度が高く、命令実行手段 111 による実行状態となり、実行中プロセスと成り得るプロセスである。

【0037】

図 2 は、本実施の形態が動作するシステム構成の例である。本実施の形態は、プロセッサ 131 と、メモリ 132 と、入出力制御部 133 と、サービスプロセッサ 134 と、周辺装置群 135 と、コンソール 136 を有している。

30

【0038】

プロセッサ 131 は、本実施の形態においては、プログラムの指示に従い演算をはじめとするデータ処理を実行するものである。命令実行手段 111、ディスパッチャ 112、タイマ 113 が、それぞれの対応するプログラムと、プロセッサ 131 によって実現されている。

【0039】

メモリ 132 は、本実施の形態においては、プログラム、および、当該プログラムのワークエリア、バッファ等を記憶するものであり、レディーキュー記憶部 114 と、割り当て時間記憶部 115 とを少なくとも有している。

40

【0040】

入出力制御部 133、サービスプロセッサ 134、周辺装置群 135、および、コンソール 136 は、一般のコンピュータシステムを構成する周知の技術であり、本実施の形態においては特別な位置づけを持たないため、説明を省略する。

【0041】

図 4 は、本発明が本実施の形態を用いて適用されるシステムにおける、プロセス識別番号 101 の構造を示す図である。ジョブ識別番号 102 は、各ジョブに対応したシステム内でユニークな番号である。ジョブ内のプロセス識別番号 103 は、各プロセスに対応し

50

たジョブ内でユニークな番号である。システム内のプロセス識別番号 101 は、ジョブ識別番号 102 とジョブ内のプロセス識別番号 103 を組み合わせた番号であり、各プロセスに対応したシステム内でユニークな番号である。本発明が本実施の形態を用いて適用されるシステムにおいて実行される各ジョブおよび各プロセスは、システム内のプロセス識別番号 101、ジョブ識別番号 102、ジョブ内のプロセス識別番号 103 で識別され、管理および制御される。

【0042】

次に、図 1 ~ 図 11 を参照して、本発明の第 1 の実施の形態の動作について、具体的な実施例を示して詳細に説明する。

【0043】

図 5 は、本発明が本実施の形態を用いて適用されるシステムにおける、プロセッサ優先割り当て時間算出の基となる周期 201 と、システム割り当て割合 202 ~ 207 との例を示したものである。周期 201 は、システム割り当てを繰り返す一定時間の周期である。システム割り当て割合 202 ~ 207 は、それぞれジョブ A ~ F 毎に対応し、周期 201 内での各ジョブへのシステム割り当て割合を示すものである。例えば、周期を 500 μ S、ジョブ A に対するシステム割り当て割合 202 を 20% とすると、ジョブ A の優先期間であるプロセッサ優先割り当て時間は、 $500 \mu S \times 0.2 = 100 \mu S$ である。

【0044】

図 6 は、本発明が本実施の形態を用いて適用されるシステムにおける、各ジョブに対する命令実行手段 111 の割り当てと、ディスパッチャ 112、タイマ 113 の関係を示したイメージである。

【0045】

ディスパッチャ 112 は、割り当て時間記憶部 115 の割り当て時間テーブル 402 をジョブ識別番号 102 に基づいて参照し、ジョブ識別番号 102 に対応する割り当て時間を取得する。ディスパッチャ 112 は、取得した割り当て時間を、タイマ 113 に出力する。

【0046】

タイマ 113 は、割り当て時間を取得して、ジョブ識別番号 102 に対応するジョブの優先期間の計時を開始する。タイマ 113 は、優先期間の計時が割り当て時間に達すると、ディスパッチャ 112 に優先ジョブの切替指示 405 を出力する。ディスパッチャ 112 は、切り替え指示 405 を取得すると、優先ジョブを切り替える。即ち、切り替え指示 405 は、命令実行手段 111 により実行すべき実行可能プロセスの切替指示となる。

【0047】

実行期間 406 ~ 419 は、対応するジョブが優先期間となり命令実行手段 111 を割り当てられていることを表示している。実行期間 406 ~ 411 では、ジョブ A ~ F の割り当て時間に対応して命令実行手段 111 が割り当てられている。ジョブ A の実行期間 412 と 414 の間に最優先ジョブの実行期間 413 が存在するが、これは実行期間 412 中において、最優先ジョブのレディーキュー 310 に実行可能プロセスが繋がられ、その先頭の実行可能プロセスが最優先で、命令実行手段 111 を割り当てられている状態を表現したものである。レディーキュー 310 に繋がられた実行可能プロセスが存在しなくなると（レディーキュー 310 が空になると）、中断されたジョブ A の実行期間 412 の続きとして、実行期間 414 となる。実行期間 414 ~ 417 までは、ジョブ A ~ D の割り当て時間に対応して命令実行手段 111 が割り当てられている。実行期間 417 と 418 の間にジョブ E の実行期間が存在しないが、これはこの時点で、ジョブ E のレディーキュー 308 が空であることを表現している。ジョブ F の実行期間 418 が終了すると、ジョブ A の実行期間 419 になる。

【0048】

図 7 は、本発明が本実施の形態を用いて適用されるシステムにおける、ジョブへの資源割当の基本の動作を示すフローチャートである。

【0049】

以下の説明において、命令実行手段 1 1 1 の状態は、命令実行手段 1 1 1 による実行状態にあるプロセスである実行中プロセスが、命令実行手段 1 1 1 上にない状態（空き状態）と、命令実行手段 1 1 1 上に取り込んで、命令実行手段 1 1 1 上に存在している状態（プロセス状態）がある。プロセス状態では、一つだけ命令実行手段 1 1 1 上に存在している実行中プロセスを実行する。命令実行手段 1 1 1 に取り込む実行中プロセスは、レディーキュー群 3 0 3 に繋がれた実行可能プロセス群 3 1 1 の中から、その時点で最優先の実行可能プロセスが、ディスパッチャ 1 1 2 により一つだけ選択される。

【 0 0 5 0 】

以下の説明において、レディーキュー m、レディーキュー n は、レディーキュー 3 0 4 ~ レディーキュー 3 0 9 のいずれかのレディーキューであることを示す。

10

【 0 0 5 1 】

ディスパッチャ 1 1 2 は、命令実行手段 1 1 1 上に最優先ジョブに属する実行中プロセスが存在するか否かを確認し、存在すればステップ 5 1 1 に分岐し、否であればステップ 5 0 2 に進む（ステップ 5 0 1 ）。

【 0 0 5 2 】

ディスパッチャ 1 1 2 は、最優先ジョブのレディーキュー 3 1 0 に実行可能プロセスが存在するか否かを確認し、存在すればステップ 5 1 4 に分岐し、否であればステップ 5 0 3 に進む（ステップ 5 0 2 ）。

【 0 0 5 3 】

ディスパッチャ 1 1 2 は、タイマ 1 1 3 から優先ジョブの切り替え指示 4 0 5 があるかどうかを確認する。処理は、優先ジョブの切り替え指示 4 0 5 が存在すればステップ 5 1 8 に分岐し、否であればステップ 5 0 4 に進む（ステップ 5 0 3 ）。

20

【 0 0 5 4 】

ディスパッチャ 1 1 2 は、命令実行手段 1 1 1 上に実行中プロセスが存在するか否かを確認し、存在すればステップ 5 0 5 に進み、否であればステップ 5 2 3 に分岐する（ステップ 5 0 4 ）。

【 0 0 5 5 】

ディスパッチャ 1 1 2 は、同一のジョブ内でプロセススイッチ要因があるか確認する。プロセススイッチ要因があればステップ 5 0 6 に進み、否であればステップ 5 0 7 に進む（ステップ 5 0 5 ）。プロセススイッチとは、命令実行手段 1 1 1 上の実行中プロセスを実行可能プロセス群 3 1 1 の中に戻し、実行可能プロセス群 3 1 1 の中から規定に従った一つの実行可能プロセスを選択し、命令実行手段 1 1 1 上の実行中プロセスとする処理を言う。

30

【 0 0 5 6 】

ディスパッチャ 1 1 2 は、命令実行手段 1 1 1 上の実行中プロセスを対応するレディーキュー m に戻すことにより実行可能プロセスとし、当該レディーキュー m の先頭に存在する実行可能プロセスを命令実行手段 1 1 1 上に取り込み実行中プロセスとする。これにより、同一のジョブ内でのプロセススイッチが実施され、ステップ 5 0 7 に進む（ステップ 5 0 6 ）。

【 0 0 5 7 】

40

命令実行手段 1 1 1 は、命令実行手段 1 1 1 上の実行中プロセスの命令を一つ実行する。この命令はプロセスを構成する命令群のひとつであり、予め定められた規則に従って命令群の中から選択された命令である。その後、処理は、ステップ 5 0 8 に進む（ステップ 5 0 7 ）。

【 0 0 5 8 】

タイマ 1 1 3 は、命令実行手段 1 1 1 上の実行中プロセスが属するジョブの計時値を更新し、当該ジョブに与えられた割り当て時間を計測する。その後、処理は、ステップ 5 0 9 に進む（ステップ 5 0 8 ）。

【 0 0 5 9 】

ディスパッチャ 1 1 2 は、命令実行手段 1 1 1 上の実行中プロセスは継続して実行可能

50

か否かを確認する。ステップ507の命令の実行結果によっては、該実行中プロセスが、実行可能状態から他の状態に遷移する規則になっているためである。処理は、継続して実行可能であればステップ501に分岐し、否であればステップ510に進む（ステップ509）。

【0060】

ディスパッチャ112は、命令実行手段111上の実行中プロセスを、実行中の状態から規則により規定された適当な状態に遷移させる。これにより、命令実行手段111上には実行中プロセスが存在しなくなる。その後、処理は、ステップ501に分岐する（ステップ510）。

【0061】

図8は、命令実行手段111上で最優先ジョブに属する実行中プロセスが存在した場合の動作を示すフローチャートである。ディスパッチャ112は、最優先ジョブ内でプロセススイッチ要因があるか確認する。処理は、要因があればステップ512に進み、否であればステップ513に進む（ステップ511）。

【0062】

ディスパッチャ112は、命令実行手段111上の実行中プロセスを最優先ジョブのレディーキュー310に戻すことにより実行可能プロセスとし、同レディーキュー310の先頭に存在する実行可能プロセスを命令実行手段111上に取り込み、実行中プロセスとする（ステップ512）。

【0063】

ディスパッチャ112は、命令実行手段111上の実行中プロセスの一つの命令を実行する。本ステップ513での実行中プロセスは最優先ジョブに属している。その後、処理は、ステップ509に進む。最優先ジョブにはプロセッサ優先割り当て時間に相当するものが無いため、ステップ508は実行しない（ステップ513）。

【0064】

図9は、命令実行手段111上に最優先ジョブに属する実行中プロセスは存在せず、かつ最優先ジョブのレディーキュー310には実行可能プロセスが存在する場合の動作を示すフローチャートである。ディスパッチャ112は、命令実行手段111上にジョブA～Fに属する実行中プロセスが存在するか否かを確認し、存在すれば、ステップ515に進み、否であればステップ516に進む（ステップ514）。

【0065】

ディスパッチャ112は、命令実行手段111上のジョブA～Fに属する実行中プロセスを対応するレディーキューmに戻し実行可能プロセスとする。その後、処理は、ステップ516に進む（ステップ515）。

【0066】

ディスパッチャ112は、最優先ジョブのレディーキュー310の先頭に存在する実行可能プロセスを命令実行手段111上に取り込み、実行中プロセスとする（ステップ516）。

【0067】

ディスパッチャ112は、ステップ516で取り込んだ実行中プロセスの一つの命令を実行する。その後、処理は、ステップ509に分岐する。最優先ジョブにはプロセッサ優先割り当て時間に相当するものが無いため、ステップ508は実行しない（ステップ517）。

【0068】

図10は、最優先ジョブに属する実行中プロセスも実行可能プロセスも存在せず、かつタイマ113から優先期間の切り替え指示405がある場合の動作を示すフローチャートである。ディスパッチャ112は、命令実行手段111上に実行中プロセスが存在するか否かを確認する。この実行中プロセスはジョブA～Fの何れかに属しているものである。処理は、命令実行手段111上に実行中プロセスが存在すればステップ519に進み、否であればステップ520に進む（ステップ518）。

10

20

30

40

50

【 0 0 6 9 】

ディスパッチャ 1 1 2 は、命令実行手段 1 1 1 上の実行中プロセスを対応するレディーキュー m に戻す。この実行中プロセスは、優先期間が終了したジョブに属したものである。その後、処理は、ステップ 5 2 0 に進む（ステップ 5 1 9 ）。

【 0 0 7 0 】

ディスパッチャ 1 1 2 は、優先期間を次のジョブに切り替え、当該ジョブに対応した割り当て時間をタイマ 1 1 3 に出力する。対応する優先期間の優先ジョブの切替指示 4 0 5 はリセットされる。その後、処理は、ステップ 5 2 1 に進む（ステップ 5 2 0 ）。

【 0 0 7 1 】

ディスパッチャ 1 1 2 は、優先期間の切替先のジョブに対応したレディーキュー n（n = 3 0 4 ~ 3 0 9 ）に実行可能プロセスが存在するか否かを確認し、存在すればステップ 5 2 2 に進み、否であればステップ 5 0 1 に分岐する（ステップ 5 2 1 ）。

10

【 0 0 7 2 】

ディスパッチャ 1 1 2 は、レディーキュー n の先頭に存在する実行可能プロセスを命令実行手段 1 1 1 上に取り込み、実行中プロセスとする。その後、処理は、ステップ 5 0 7 に分岐する（ステップ 5 2 2 ）。

【 0 0 7 3 】

図 1 1 は、最優先ジョブに属する実行中プロセスおよび実行可能プロセスが存在せず、タイマ 1 1 3 から優先期間の切り替え指示 4 0 5 も無く、かつ、命令実行手段 1 1 1 上にジョブ A ~ F に属する実行中プロセスも無い場合の動作を示すフローチャートである。ディスパッチャ 1 1 2 は、優先期間をアサインされたジョブに対応したレディーキュー m に実行可能プロセスが存在するか否かを確認し、存在する場合はステップ 5 2 4 に進む。否であれば、処理は、ステップ 5 2 0 に分岐し、優先期間を次のジョブに切り替える（ステップ 5 2 3 ）。

20

【 0 0 7 4 】

ディスパッチャ 1 1 2 は、優先期間をアサインされたジョブに対応したレディーキュー m の先頭に存在する実行可能プロセスを命令実行手段 1 1 1 上に取り込み、実行中プロセスとする。その後、処理は、ステップ 5 0 7 に分岐する（ステップ 5 2 4 ）。

【 0 0 7 5 】

このようにして、ディスパッチャ 1 1 2 は、上記に述べた規則に従い、あるジョブの優先割り当て時間内で、規則に従い、当該ジョブのに対応するレディーキューの先頭から当該ジョブに属するプロセスを取り込み、命令実行手段 1 1 1 上の実行中プロセスとする。

30

【 0 0 7 6 】

本発明の第 1 の実施の形態によれば、あらかじめ定められたジョブ毎のシステム割り当て割合に基づいたプロセッサ割り当て時間をジョブ毎の優先期間とし、ジョブ毎に対応するレディーキューには、各ジョブに属する実行可能プロセスを繋ぎ、ジョブ毎の優先期間において、ジョブ毎に対応するレディーキューの先頭に繋がれた実行可能プロセスを選択し、プロセッサに割り当て実行させることができるようにしたことにより、各ジョブに対するプロセッサの割り当て時間は、予め設定された各ジョブへのシステム割り当て割合とほぼ同じになり、複数のジョブを単一のシステムで実行する場合において、特定のジョブの負荷が増大した場合でも、他のジョブが実行出来ない状況やシステムの割り当て割合が減少する状況を防止することが可能になるという効果を有している。

40

【 0 0 7 7 】

また、本発明の第 1 の実施の形態によれば、一つのジョブが、優先ジョブの期間であっても、実行すべき仕事が無くなり、優先ジョブのレディーキューが空であった場合、他のジョブが優先ジョブに切り替わるようにしたことにより、特定のジョブの負荷が軽くなり、ジョブのシステム使用時間が予め定めたシステム割り当て割合に満たない場合は、システム時間を無駄にすることなく、他のジョブにシステムを割り当てることが可能になるという効果を有している。

【 0 0 7 8 】

50

さらに、本発明の第 1 の実施の形態によれば、システム共通処理や緊急性を要するジョブを最優先ジョブと定義し、最優先ジョブに対応するレディーキューを備え、レディーキューに実行可能プロセスが繋がれた場合、即ちこの最優先ジョブで実行すべき仕事が発生すると、最優先でシステムを割り当てるようにしたことにより、最優先ジョブについては、一般のジョブが優先ジョブである期間か否かに関わらず、最優先でシステムを割り当てるのが可能になるという効果を有している。

【0079】

次に本発明の第 2 の実施の形態について図面を参照して詳細に説明する。

【0080】

図 12 は、本発明の第 2 の実施の形態の構成を示すブロック図である。本発明の第 2 の実施の形態においては、本発明の第 1 の実施の形態の構成を示すブロック図である図 1 に対して、割り当て時間算出手段 116 が追加されていることが、本発明の第 1 の実施の形態と異なる。

【0081】

割り当て時間算出手段 116 は、例えば図 2 に示すサービスプロセッサ 134 や、コンソール 136 から入力された、図 5 に示す周期 201 と、ジョブ毎のシステム割り当て割合 202 ~ 207 を取得する。そして、割り当て時間算出手段 116 は、図 5 に示す周期 201 と、ジョブ毎のシステム割り当て割合 202 ~ 207 に基づいて、ジョブ毎のプロセッサ優先割り当て時間を算出し、図 6 割り当て時間テーブル 402 に格納する。

【0082】

第 2 の実施の形態の動作について、図 5 に示す例により、具体的な実施例を詳細に説明する。割り当て時間算出手段 116 は、図示しない何らかの手段から、図 5 に示す周期 $201 = T [\mu s]$ として『500』を、ジョブ A のシステム割り当て割合 $202 = R_a [\%]$ として『20』を、ジョブ B のシステム割り当て割合 $203 = R_b [\%]$ として『10』、ジョブ C のシステム割り当て割合 $204 = R_c [\%]$ として『30』を、ジョブ D のシステム割り当て割合 $205 = R_d [\%]$ として『12』を、ジョブ E のシステム割り当て割合 $206 = R_e [\%]$ として『18』を、ジョブ F のシステム割り当て割合 $207 = R_f [\%]$ として『10』、を取得する。そして、割り当て時間算出手段 116 は、これらに基づいて、ジョブ A のプロセッサ優先割り当て時間 $= 500 [\mu s] \times 0.2 = 100 [\mu s]$ を、ジョブ B のプロセッサ優先割り当て時間 $= 500 [\mu s] \times 0.1 = 50 [\mu s]$ を、ジョブ C のプロセッサ優先割り当て時間 $= 500 [\mu s] \times 0.3 = 150 [\mu s]$ を、ジョブ D のプロセッサ優先割り当て時間 $= 500 [\mu s] \times 0.12 = 60 [\mu s]$ を、ジョブ E のプロセッサ優先割り当て時間 $= 500 [\mu s] \times 0.18 = 90 [\mu s]$ を、ジョブ F のプロセッサ優先割り当て時間 $= 500 [\mu s] \times 0.1 = 100 [\mu s]$ を算出する。そして、割り当て時間算出手段 116 は、図 6 に示す割り当て時間テーブル 402 に格納する。

【0083】

さらに、割り当て時間算出手段 116 は、ジョブの終了、ジョブの追加が発生した場合にプロセッサ優先割り当て時間を再算出するための再算出機能を有する。

【0084】

以下に、図 5 に示す状態に対して、ジョブ G が追加された場合の、プロセッサ優先割り当て時間を再算出する動作について具体的に説明する。ディスパッチャ 112 は、ジョブの追加を検出すると割り当て時間算出手段 116 にプロセッサ優先割り当て時間の再算出を指示する。割り当て時間算出手段 116 は、ジョブ G のシステム割り当て割合 $R_g [\%] = 『10』$ を取得する。ここで、図 5 に示す周期 $201 = 『500 [\mu s]』$ は、既存のジョブ A ~ F のシステム割り当て割合の合計『100%』で配分されている。このため、割り当て時間算出手段 116 は、ジョブ G のシステム割り当て割合を合わせて、全体が 100% になるようにジョブ A ~ F のシステム割り当て割合を再計算する。割り当て時間算出手段 116 は、ジョブ A のシステム割り当て割合 $202 = R_a [\%] = 20 [\%] \div 1.1 = 18 [\%]$ 、ジョブ B のシステム割り当て割合 $203 = R_b [\%] = 10 [\%]$

10

20

30

40

50

$\div 1.1 = 9 [\%]$ 、ジョブ C 以下も同様であり、ジョブ G のシステム割り当て割合 $R_g [\%] = 10 [\%] \div 1.1 = 9 [\%]$ を算出する。

【0085】

そして、割り当て時間算出手段 116 は、この再計算した各ジョブのシステム割り当て割合に基づいて、各ジョブのプロセッサ優先割り当て時間を再算出する。

【0086】

尚、以上の例では、周期を変えずにシステム割り当て割合を再計算する場合を説明したが、周期 $201 = \text{『} 500 [\mu s] \text{』}$ に対して、ジョブ G のシステム割り当て割合 $R_g [\%] = \text{『} 10 \text{』}$ に当たる分のプロセッサ優先割り当て時間 $\text{『} 50 \mu s \text{』}$ を周期に追加して周期 $201 = \text{『} 550 [\mu s] \text{』}$ として再算出しても良い。

10

【0087】

また、ジョブが終了した場合も、容易に同様の方法を適用できる。

【0088】

本発明の第 2 の実施の形態によれば、割り当ての周期とジョブ毎のシステム割り当て割合に基づいて、また、ジョブの終了、追加の発生に対応してジョブ毎のプロセッサ優先割り当て時間を算出するようにしたことにより、プロセッサ優先割り当て時間をあらかじめ計算する手間を省略できる汎用的な、ジョブへの資源割当方式を提供することが可能になるという効果を有している。

【産業上の利用可能性】

【0089】

20

本発明は、一般の情報処理装置で活用可能であるが、特に一つのプロセッサ上で複数の機能を処理する場合などに有効である。

【0090】

本発明の活用例として、一つのプロセッサ上において、ネットワーク、ディスク、テープ、コンソールなどの入出力処理を行わせる場合、各処理に対して、予め定めた割合で効率良くプロセッサを使用させることが出来る。

【図面の簡単な説明】

【0091】

【図 1】本発明の第 1 の実施の形態の構成を示すブロック図である。

【図 2】本発明の第 1 および第 2 の実施の形態のシステム構成を示す図である。

30

【図 3】本発明の第 1 の実施の形態におけるレディーキュー群および実行可能プロセス群のイメージを示す図である。

【図 4】本発明の第 1 の実施の形態におけるプロセス識別番号の構造を示す図である。

【図 5】本発明の第 1 の実施の形態における周期 201 と、システム割り当て時間との例である。

【図 6】本発明の第 1 の実施の形態における各ジョブに対するプロセッサの割り当てと、ディスパッチャ、タイマの関係のイメージを示す図である。

【図 7】本発明の第 1 の実施の形態におけるジョブへの資源割当の基本の動作を示すフローチャートである。

【図 8】本発明の第 1 の実施の形態におけるプロセッサ上で最優先ジョブに属する実行中プロセスが存在した場合の動作を示すフローチャートである。

40

【図 9】本発明の第 1 の実施の形態におけるプロセッサ上に最優先ジョブに属する実行中プロセスは存在せず、かつ最優先ジョブのレディーキューには実行可能プロセスが存在する場合の動作を示すフローチャートである。

【図 10】本発明の第 1 の実施の形態における最優先ジョブに属する実行中プロセスも実行可能プロセスも存在せず、かつタイマから優先期間の切り替え指示がある場合の動作を示すフローチャートである。

【図 11】本発明の第 1 の実施の形態における最優先ジョブに属する実行中プロセスおよび実行可能プロセスが存在せず、タイマから優先期間の切り替え指示も無く、かつ、プロセッサ上にジョブ A ~ F に属する実行中プロセスも無い場合の動作を示すフローチャート

50

である。

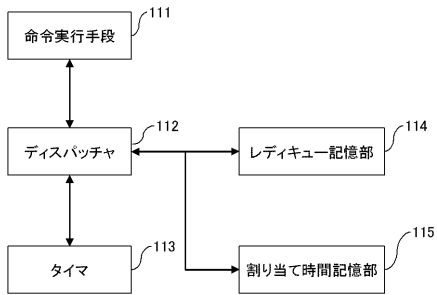
【図 1 2】本発明の第 2 の実施の形態の構成を示すブロック図である。

【符号の説明】

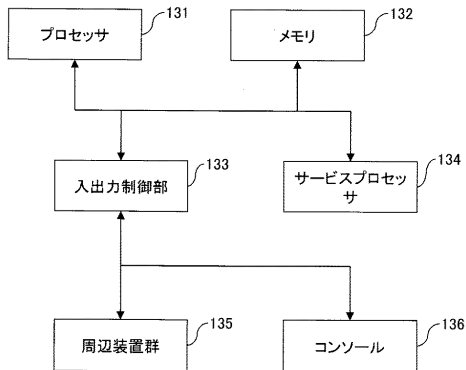
【 0 0 9 2 】

1 0 1	システム内のプロセス識別番号	
1 0 2	ジョブ識別番号	
1 0 3	ジョブ内のプロセス識別番号	
1 1 1	命令実行手段	
1 1 2	ディスパッチャ	
1 1 3	タイマ	10
1 1 4	レディーキュー記憶部	
1 1 5	割り当て時間記憶部	
1 1 6	割り当て時間算出手段	
1 3 1	プロセッサ	
1 3 2	メモリ	
1 3 3	入出力制御部	
1 3 4	サービスプロセッサ	
1 3 5	周辺装置群	
1 3 6	コンソール	
2 0 1	周期	20
2 0 2 ~ 2 0 7	システム割り当て割合	
3 0 4 ~ 3 0 9	レディーキュー	
3 1 0	レディーキュー	
3 1 1	実行可能プロセス群	
4 0 2	割り当て時間テーブル	
4 0 5	優先ジョブの切替指示	
4 0 6 ~ 4 1 9	実行期間	

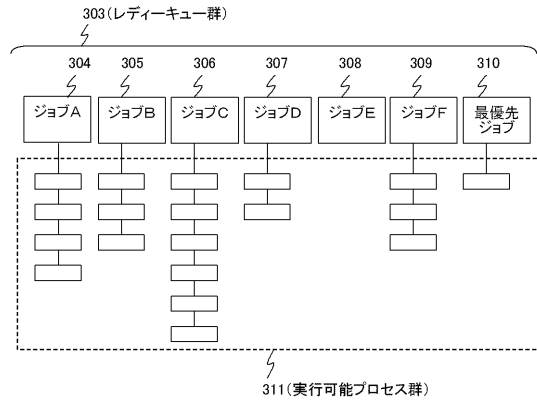
【図 1】



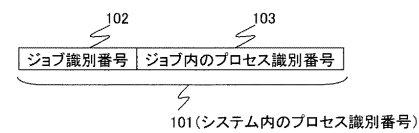
【図 2】



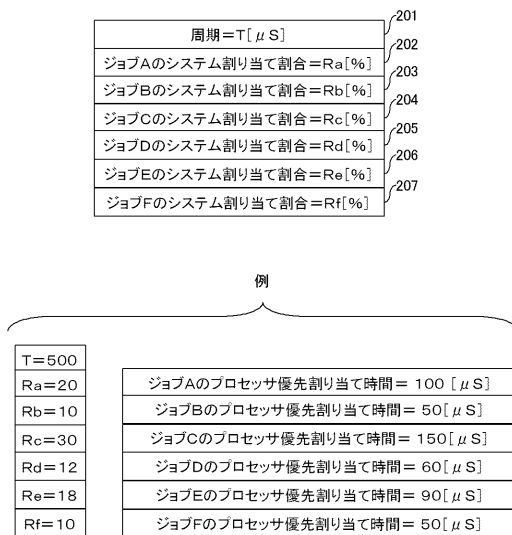
【図 3】



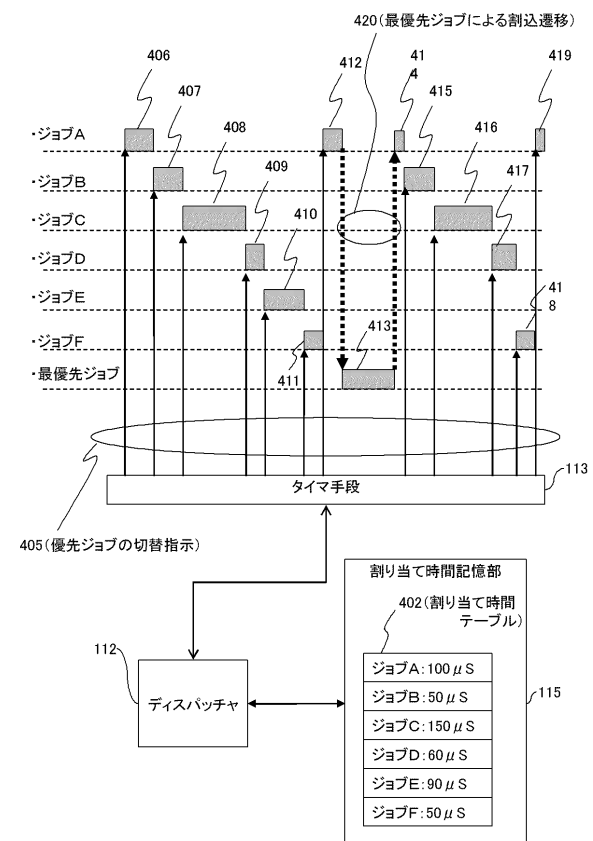
【図 4】



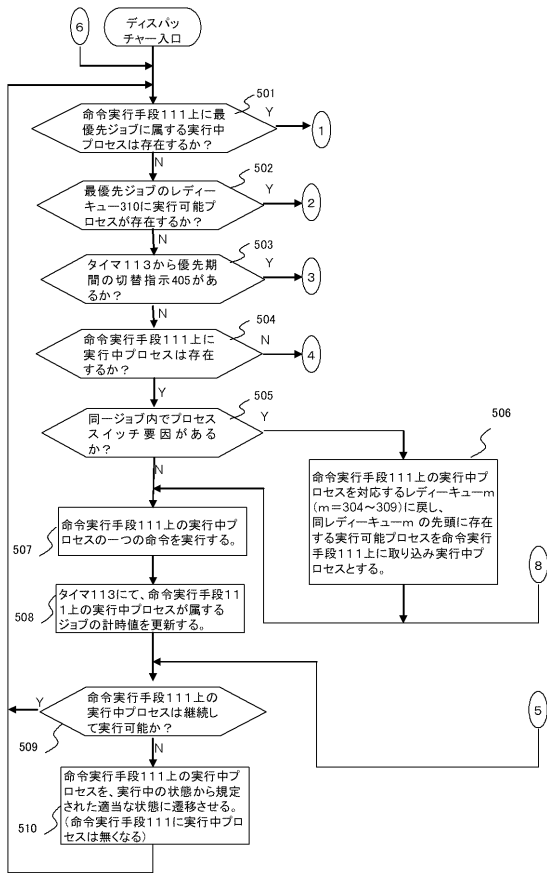
【図 5】



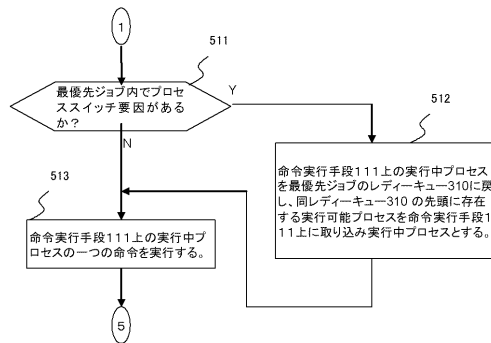
【図 6】



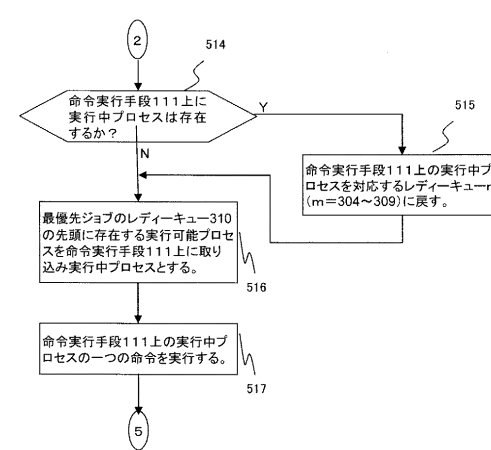
【図 7】



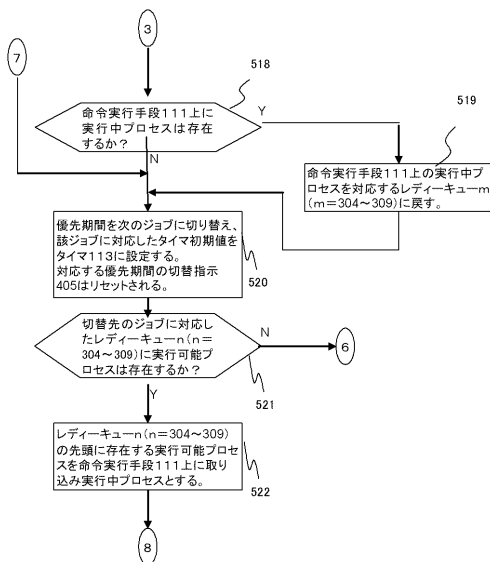
【図 8】



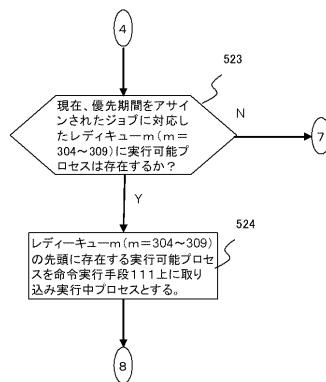
【図 9】



【図 10】



【図 11】



【図 12】

