(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第3711642号
(P3711642)

(45)発行日　平成17年11月2日(2005.11.2)　(24)登録日　平成17年8月26日(2005.8.26)

(51)Int.Cl.[7]　F I
G06F　9/48　G06F　9/46　４５２Ｂ

請求項の数 5　(全 16 頁)

(21)出願番号　特願平8-195486
(22)出願日　平成8年7月5日(1996.7.5)
(65)公開番号　特開平10-21092
(43)公開日　平成10年1月23日(1998.1.23)
審査請求日　平成14年4月17日(2002.4.17)

(73)特許権者　000005496
富士ゼロックス株式会社
東京都港区赤坂二丁目１７番２２号
(74)代理人　100098132
弁理士　守山　辰雄
(72)発明者　柴田　博仁
神奈川県足柄上郡中井町境４３０　グリーンテクなかい　富士ゼロックス株式会社内
(72)発明者　小松原　弘文
神奈川県足柄上郡中井町境４３０　グリーンテクなかい　富士ゼロックス株式会社内

審査官　後藤　彰

最終頁に続く

(54)【発明の名称】処理管理システム

(57)【特許請求の範囲】

【請求項１】

処理を実行する処理実行部を備え、指定された実行優先度に従って処理実行部が処理を実行する処理管理システムにおいて、

実行途中の２つの処理についての実行状態を管理することにより当該２つの処理間で優先度逆転が発生していることを検出する優先度逆転検出部と、

優先度逆転検出部が優先度逆転を検出したことに基づいて、前記２つの処理のうちで高い実行優先度が指定された第１処理と低い実行優先度が指定された第２処理との実行優先度の差を算出する優先度差算出部と、

算出された実行優先度差と予め設定された規定値とを比較する優先度差判定部と、

優先度差判定部により実行優先度差が規定値未満と判定されたことに基づいて、第２処理の実行優先度を第１処理の実行優先度以上にして処理実行部に第２処理を実行させる実行制御対応部と、

優先度差判定部により実行優先度差が規定値以上と判定されたことに基づいて、第２処理をアボートするアボート対応部と、を備えたことを特徴とする処理管理システム。

【請求項２】

処理を実行する処理実行部を備え、指定された実行優先度に従って処理実行部が処理を実行する処理管理システムにおいて、

前記実行優先度として指定可能な範囲は、処理実行部を占有して実行される処理に指定する高実行優先度の範囲と、処理実行部を占有せずに実行される処理に指定する低実行優

先度の範囲とを有し、
実行途中の２つの処理についての実行状態を管理することにより当該２つの処理間で優先度逆転が発生していることを検出する優先度逆転検出部と、
優先度逆転検出部が優先度逆転を検出したことに基づいて、前記２つの処理のうちで高い実行優先度が指定された第１処理と低い実行優先度が指定された第２処理との実行優先度がそれぞれ上記したいずれの範囲に属するかを判定する優先度逆転判定部と、
優先度逆転判定部により第１処理が高実行優先度の範囲に属するとともに、第２処理が低実行優先度の範囲に属すると判定されたことに基づいて第２処理をアボートする第２アボート対応部と、を備えたことを特徴とする処理管理システム。
【請求項３】
請求項２に記載した処理管理システムにおいて、
前記優先度逆転判定部により第１処理及び第２処理が共に高実行優先度の集合に属すると判定されたことに基づいて、第１処理と第２処理との実行優先度の差を算出する優先度差算出部と、
算出された実行優先度差と予め設定された規定値とを比較する優先度差判定部と、
優先度差判定部により実行優先度差が規定値未満と判定されたことに基づいて、第２処理の実行優先度を第１処理の実行優先度以上にして処理実行部に第２処理を実行させる実行制御対応部と、
優先度差判定部により実行優先度差が規定値以上と判定されたことに基づいて、第２処理をアボートするアボート対応部と、を備えたことを特徴とする処理管理システム。
【請求項４】
処理を実行する処理実行部を備え実行優先度に従って前記処理実行部により処理が実行されるコンピュータ処理管理システムにおける処理管理方法において、
実行途中の２つの処理間で優先度逆転が発生していることが検出されたことに基づいて、
高い実行優先度が指定された第１処理と低い実行優先度が指定された第２処理との実行優先度の差が算出され、
算出された実行優先度差と予め設定された規定値とが比較され、実行優先度差が規定値未満の場合には、第２処理の実行優先度が第１処理の実行優先度以上に設定されて第２処理が実行され、実行優先度差が規定値以上の場合には、第２処理がアボートされることを特徴とするコンピュータ処理管理システムにおける処理管理方法。
【請求項５】
処理を実行する処理実行部を備え実行優先度に従って前記処理実行部により処理が実行されるコンピュータ処理管理システムにおける処理管理方法において、
処理実行部を占有して実行される処理に指定される高実行優先度の範囲と、処理実行部を占有せずに実行される処理に指定される低実行優先度の範囲とが予め設定され、
実行途中の２つの処理間で優先度逆転が発生していることが検出されたことに基づいて、高い実行優先度が指定された第１処理と低い実行優先度が指定された第２処理との実行優先度がそれぞれ上記いずれの範囲に属するかが判定され、
第１処理が高実行優先度の範囲に属するとともに第２処理が低実行優先度の範囲に属する場合には、第２処理がアボートされることを特徴とするコンピュータ処理管理システムにおける処理管理方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、実行優先度に基づいて複数の処理に与える計算機資源等を管理して処理を実行する処理管理システムに関し、特に、処理間で発生する優先度逆転を解消する処理管理システム及び方法に関する。
【０００２】
【従来の技術】

処理管理システムは、ＣＰＵといった計算機資源の割り当て等を管理して複数の処理の実行を可能にする方式として、例えば、一定時間毎に計算機資源を割り当てる処理を切り換えて、複数の処理を実行するラウンドロビン方式等があり、また、ラウンドロビン方式において実行優先度に従って上記した計算機資源を割り当てる処理を決定する方式もある。

【０００３】

また、上記したラウンドロビン方式に対して、例えば、映像や音声といった連続メディア・データの再生、ユーザから時間の制約を受けたデータベースへの検索といった実時間処理等に適した実行を実現する方式として、計算機資源を実行優先度に従って占有させて処理を実行する方式（以下、従優先度占有方式という。）がある。

【０００４】

この従優先度占有方式は、処理を実行する際に実行優先度に従って計算機資源を占有させるために、高実行優先度が指定された処理を実行する際に、他の低実行優先度が指定された処理によって計算機資源を奪われることを防ぐことができ、多くの計算機資源を高実行優先度が指定された処理へ与えることができる。

【０００５】

したがって、高実行優先度が指定された処理に対して、開始から終了にかかる応答時間を短縮することができ、指定された終了期限内まで或いは早期に終了しなければ処理の価値が無効或いは低下してしまうといった性質を持った実時間処理に対して適した実行が可能となっている。

【０００６】

上記した処理管理システムにおいては、計算機資源を有効に使用してより多くの処理を実行することが望ましく、さらに、実行優先度を使用している処理管理システムにおいては、実行優先度に従った順番で処理を実行することが望ましい。

このことは、従優先度占有方式で処理を実行する処理管理システムにおいては、他方式で処理を実行する処理管理システムよりも強く望まれている。

【０００７】

このような処理管理システムにおいて、ラウンドロビン方式を実行している場合、従優先度占有方式を実行している場合等に限らず、実行優先度に従って処理を実行する場合には、高実行優先度の処理T_{high}がロックを要求したデータに対して低実行優先度の処理T_{low}がロックを確保しているために、高実行優先度の処理T_{high}のロックがブロックされるといった優先度逆転が発生し、高実行優先度の処理T_{high}の応答時間が長くなってしまうといった問題が生じていた。

【０００８】

上記した問題は従優先度占有方式においては特に致命的な問題となっている。

例えば、１つのＣＰＵを備えている場合（所謂、シングルプロセッサの場合）には、占有して行うことのできる処理の個数は１つのみであり、高実行優先度の処理T_{high}以外の処理が実行されることがなく、高実行優先度の処理T_{high}は低実行優先度の処理T_{low}にロックされているデータを使用できないために実行を進行させることができず応答時間が限りなく延びてしまうこととなる。

また、複数のＣＰＵを備えている場合（所謂、マルチプロセッサの場合）には、上記した高実行優先度の処理T_{high}の応答時間が限りなく延びてしまう事態は避けることができるものの、高実行優先度T_{high}の応答時間が長いことには変わりがない。

【０００９】

すなわち、マルチプロセッサの場合には、占有して行うことのできる処理の個数を複数とすることができ、低実行優先度の処理より高実行優先度の処理（処理T_{high}以外）の実行が終了した後に低実行優先度T_{low}の処理が実行されるので、低実行優先度の処理T_{low}が終了した後に高実行優先度の処理T_{high}の実行を進行させることができる。しかしながら、高実行優先度の処理T_{high}の実行が進行するのは、低実行優先度の処理より高実行優先度の処理（処理T_{high}以外）の実行が終了した後、さらに低実行優先度の処理T_{low}が終了した後といった長時間経過した後である。

【００１０】
前記したような優先度逆転が引き起こす問題に対する解決策として、R.Abbott and H.Garcia-Molina,Scheduling Real-Time Transactions: A Performance Evaluation,ACM Transactions on Database Systems,Vol.17,No.3,1992.（以下、文献１という。）には、低実行優先度の処理T_{low}をアボート（無効化）し、高実行優先度の処理T_{high}によるデータへのロック確保を可能にして処理を進行させるＨＰ(High Priority)方式が提案されている。
【００１１】
このＨＰ方式は、高実行優先度の処理を最優先に実行させるといった点に関しては優れた方式ではあるが、複数の処理をできるだけ効率よく実行するといった点においては優れたものということができなかった。すなわち、低実行優先度の処理をアボートするために、低実行優先度の処理に与えた計算機資源を無駄にしてしまっており、複数の処理を実行する場合において処理に対する実行効率を低下させていた。
【００１２】
上記したＨＰ方式に対して、複数の処理に対する実行効率を向上させる方式として、文献１にはＷＰ(Wait Promote)方式が記載されている。
このＷＰ方式は計算機資源の無駄を削減することを考えた方式であり、高実行優先度の処理T_{high}と低実行優先度の処理T_{low}との間で優先度逆転が発生した際に、処理T_{low}の実行優先度を処理T_{high}と同じ実行優先度に引き上げることによって、処理T_{low}を実行させ、その後に処理T_{high}を実行させる。したがって、処理T_{low}をアボートする等の計算機資源の無駄を減少させることができる。
【００１３】
さらに、低実行優先度の処理T_{low}の実行優先度を高実行優先度の処理T_{high}と同じ実行優先度へ引き上げるために、処理T_{low}が実行されて終了した後に、直ちに処理T_{high}が実行されることとなり、例えば、上記説明したマルチプロセッサにおいて発生していた、処理T_{high}より実行優先度の低い処理（処理T_{low}以外）が終了すること待たなければ処理T_{high}を実行できなかったといった状況を防ぐことができ、処理T_{high}の応答時間を短縮をすることができる。
【００１４】
また、文献１には、実行優先度の高い処理を期限内に終了させることを目的としてＷＰ方式及びＨＰ方式を組み合わせたＣＲ(Conditional Restart)方式が提案されている。
このＣＲ方式は、高実行優先度の処理T_{high}と低実行優先度の処理T_{low}との間で優先度逆転が発生した場合に、ＷＰ方式を適用しても処理T_{high}が期限内に終了するか否かを判断し、終了期限内に終了させることができるときにはＷＰ方式を適用し、終了させることができないときにはＨＰ方式を適用する方式である。
【００１５】
【発明が解決しようとする課題】
上記したように優先度逆転が発生した場合の対応策として、ＨＰ方式、ＷＰ方式、ＣＲ方式等がある。
しかしながら、上記した各方式は処理管理システムにおいて複数の処理を実行することに適したものではなかった。
例えば、ＨＰ方式は実行優先度の順番に処理を実行するといった点では優れた方式であるものの、処理をアボートするといった計算機資源を無駄にしており、効率よく処理を実行することができなかった。
また、ＷＰ方式は、処理をアボートするといった計算機資源の無駄を削減し、処理を効率よく実行するといった点では優れた方式であるものの、実行優先度の順番に従って処理を実行することはできなかった。
【００１６】
また、ＣＲ方式はＨＰ方式及びＷＰ方式を選択的に利用して両者の利点を併せ持つことによって処理を期限内に終了させるといった点では優れた方式であるものの、このＣＲ方式

を利用するためには、処理にかかる実行時間を予め把握しておくことが必要であり、例えば、処理量を確定することのできないデータベースへの検索等といった処理実行前に実行時間を把握することができない処理に対しては、適用できず何ら効果を得ることができなかった。

【００１７】

そこで、本発明は上記した従来の事情に鑑みなされたもので、処理にかかる実行時間の分かっていない場合でも、優先度逆転への対応としてＷＰ方式とＨＰ方式とを選択して適用する等でき、実行優先度にできるだけ忠実に処理を実行するとともに、計算機資源の無駄を省いて効率よく処理を実行することのできる処理管理システム及び方法を提供することを目的としている。

【００１８】

【課題を解決するための手段】

上記目的を達成するために本発明は、２つの処理間の実行優先度の差が比較的小さいときには、前記処理を実行する順番に対する制約は弱いものであり、また、２つの処理間の実行優先度の差が比較的大きいときには、前記実行を実行する順番に対する制約が強いものであることに着目してなされたもので、優先度逆転が生じたときに、優先度逆転が起きている２つの処理間の実行優先度差に基づいて、優先度逆転への対応方式を選択して適用する。

すなわち、実行優先度差が比較的小さいといった処理に対する実行順番の制約が弱いときには、計算機資源の無駄を防ぐＷＰ方式を適用し、実行優先度差が比較的大きいといった実行優先度の制約が強いときにはＨＰ方式を適用する。

【００１９】

さらに、本発明は、比較的高い実行優先度が指定される計算機資源を占有して実行される占有処理と、比較的低い実行優先度が指定される計算機資源を占有せずに実行される非占有処理との間には、処理を実行する順番に対する制約が強いことに着目してなされたもので、占有処理と非占有処理との間に優先度逆転が生じたときには、非占有処理をアボートして占有処理を実行する。

【００２０】

そこで、請求項１に係る処理管理システムは、処理を実行する処理実行部を備え、指定された実行優先度に従って処理実行部が処理を実行する処理管理システムにおいて、実行途中の２つの処理についての実行状態を管理することにより当該２つの処理間で優先度逆転が発生していることを検出する優先度逆転検出部と、

優先度逆転検出部が優先度逆転を検出したことに基づいて、前記２つの処理のうちで高い実行優先度が指定された第１処理と低い実行優先度が指定された第２処理との実行優先度の差を算出する優先度差算出部と、算出された実行優先度差と予め設定された規定値とを比較する優先度差判定部と、優先度差判定部により実行優先度差が規定値未満と判定されたことに基づいて、第２処理の実行優先度を第１処理の実行優先度以上にして処理実行部に第２処理を実行させる実行制御対応部と、優先度差判定部により実行優先度差が規定値以上と判定されたことに基づいて、第２処理をアボートするアボート対応部と、を備えたことを特徴とする。

【００２１】

上記した請求項１に係る処理管理システムでは、優先度逆転検出部が実行途中の２つの処理についての実行状態を管理することにより当該２つの処理間で優先度逆転が発生していることを検出し、検出したことに基づいて、優先度算出部が比較的高い実行優先度が指定された第１処理と比較的低い実行優先度が指定された第２処理との実行優先度の差を算出する。次いで、優先度差判定部が算出された実行優先度差と予め設定された規定値とを比較し、実行優先度差が規定値未満と判定された場合には、実行制御対応部が第２処理の実行優先度を第１処理の実行優先度以上にして処理実行部に第２処理を実行させる一方、実行優先度差が規定値以上と判定された場合には、アボート対応部が第２処理をアボートする。

【００２２】
請求項２に係る処理管理システムは、処理を実行する処理実行部を備え、指定された実行優先度に従って処理実行部が処理を実行する処理管理システムにおいて、前記実行優先度として指定可能な範囲は、処理実行部を占有して実行される処理に指定する高実行優先度の範囲と、処理実行部を占有せずに実行される処理に指定する低実行優先度の範囲とを有し、実行途中の２つの処理についての実行状態を管理することにより当該２つの処理間で優先度逆転が発生していることを検出する優先度逆転検出部と、優先度逆転検出部が優先度逆転を検出したことに基づいて、前記２つの処理のうちで高い実行優先度が指定された第１処理と低い実行優先度が指定された第２処理との実行優先度がそれぞれ上記したいずれの範囲に属するかを判定する優先度逆転判定部と、優先度逆転判定部により第１処理が高実行優先度の範囲に属するとともに、第２処理が低実行優先度の範囲に属すると判定されたことに基づいて第２処理をアボートする第２アボート対応部と、を備えたことを特徴とする。

【００２３】
上記した請求項２に係る処理管理システムでは、優先度逆転検出部が実行途中の２つの処理についての実行状態を管理することにより当該２つの処理間で優先度逆転が発生していることを検出する。この検出に基づいて、優先度逆転判定部が比較的高い実行優先度が指定された第１処理と比較的低い実行優先度が指定された第２処理との実行優先度がそれぞれ、処理実行部を占有して実行される処理に指定する高実行優先度の範囲と、処理実行部を占有せずに実行される処理に指定する低実行優先度の範囲とのいずれの範囲に属するかを判定する。そして、優先度逆転判定部により第１処理が高実行優先度の範囲に属するとともに、第２処理が低実行優先度の範囲に属すると判定された場合には、第２アボート対応部が第２処理をアボートする。

【００２４】
請求項３に係る処理管理システムは、請求項２に記載した処理管理システムにおいて、前記優先度逆転判定部により第１処理及び第２処理が共に高実行優先度の集合に属すると判定されたことに基づいて、第１処理と第２処理との実行優先度の差を算出する優先度差算出部と、算出された実行優先度差と予め設定された規定値とを比較する優先度差判定部と、優先度差判定部により実行優先度差が規定値未満と判定されたことに基づいて、第２処理の実行優先度を第１処理の実行優先度以上にして処理実行部に第２処理を実行させる実行制御対応部と、優先度差判定部により実行優先度差が規定値以上と判定されたことに基づいて、第２処理をアボートするアボート対応部と、を備えたことを特徴とする。

【００２５】
上記した請求項３に係る処理管理システムでは、優先度逆転判定部により第１処理及び第２処理が共に高実行優先度の集合に属すると判定された場合に、優先度差算出部が第１処理と第２処理との実行優先度の差を算出し、優先度差判定部が算出された実行優先度差と予め設定された規定値とを比較する。そして、優先度差判定部により実行優先度差が規定値未満と判定された場合には、実行制御対応部が第２処理の実行優先度を第１処理の実行優先度以上にして処理実行部に第２処理を実行させる一方、実行優先度差が規定値以上と判定された場合には、アボート対応部が第２処理をアボートする。
また、請求項４に係る処理管理方法及び請求項５に係る処理管理方法を提供する。

【００２６】
【発明の実施の形態】
本発明の第１実施例に係る処理管理システムを図１を参照して説明する。
この処理管理システムは、処理の要求を受け付ける処理要求受け付け部１と、処理に対する実行優先度を指定する優先度決定部２と、処理に関する情報を保持する処理情報管理テーブル３と、処理情報管理テーブルに保持させる情報を管理する処理情報管理部４と、処理を実行するオペレーティングシステム５（以下、ＯＳという。）と、実行途中の処理に優先度逆転が発生したことを検出する優先度逆転検出部６と、優先度逆転の発生に対する対応を行う優先度逆転対応部７と、実行中の処理が終了期限を超えて実行されているオー

バーロード状態を検出するオーバーロード検出部８と、オーバーロード状態となった処理に対して対応を行うオーバーロード対応部９と、処理の終了を行う終了処理部１０と、を備えている。

【００２７】

処理要求受け付け部１は、実行要求のあった処理を複数格納する待ち行列を有しており、定期的にＯＳ５に割込をかけて起動され、処理要求を受け取った場合には、受け取った処理を順次待ち行列に格納し、格納している処理を順次優先度決定部２へ渡す。

【００２８】

優先度決定部２は、処理要求受け付け部１から受け取った処理に基づいて当該処理に対する実行優先度を指定し、当該処理とともに指定した実行優先度を処理情報管理部４に渡す。

なお、実行優先度は、ユーザからの指示、処理の実行にかかる時間の長さ、処理を要求したユーザに与えられている優先度、処理を要求したターミナルに与えられている優先度、処理に与えられた期限等といった基準のいずれか一つ或いは複数に基づいて指定される。

【００２９】

処理情報管理テーブル３は、処理に関する情報を保持しており、保持する情報には、例えば、図２に示すように処理の識別番号（処理ＩＤ）、処理の終了期限、処理に対して指定された実行優先度等がある。

【００３０】

処理情報管理部４は、優先度決定部２から処理を受け取ると起動され、当該処理に関する情報を処理情報管理テーブル３に設定し、当該処理をＯＳ５に渡す。

すなわち、優先度決定部２から処理を受け取って起動されると、処理情報管理部４は受け取った処理に関する情報のテーブルを新たに処理情報管理テーブル３に追加し、追加したテーブルへ当該処理の処理ＩＤ、処理の終了期限、指定された実行優先度を設定し、当該処理をＯＳ５に渡す。

次いで、処理情報管理部４は、他の処理が実行要求受け付け部１に格納されているか否かをチェックして、処理が格納されていない場合にはＯＳ５に制御を移す一方、処理が格納されている場合には処理要求受け付け部１に制御を移す。

【００３１】

ＯＳ５は、処理情報管理部４から渡された処理を格納する実行待ち処理保持部５ａと、処理に対して実行権を与える実行処理決定部５ｂと、実行権が与えられた処理を実行する処理実行部５ｃと、新たな処理に実行権を奪われた処理等を格納する中断処理保持部５ｄと、を有している。

【００３２】

実行待ち処理保持部５ａは、処理を一時格納させるための待ち行列を備え、当該待ち行列には実行優先度が設定され実行待ち状態にある処理が格納され、実行優先度の高い処理から順に並んだ状態となっている。

【００３３】

実行処理決定部５ｂは、実行待ち処理保持部５ａに新たな処理が格納された直後、処理実行部５ｃが実行している処理終了した直後、若しくは優先度逆転対応部７による指示があった直後に起動され、実行待ち処理保持部５ａに格納されている処理、処理実行部６ｃで実行されている処理、中断処理保持部５ｄに格納されている処理、から最も実行優先度の高い処理を検出し、その処理に実行権を与える。

【００３４】

例えば、実行待ち処理保持部５ａに格納されている処理へ実行権が与えられた場合には、当該処理を実行待ち処理保持部５ａの待ち行列から取り出して、処理実行部５ｃに渡す。この時に処理実行部５ｃが処理を実行中であれば、実行中の処理を中断処理保持部５ｄに格納する。

また、中断処理保持部５ｄに格納されている処理へ実行権が与えられた場合には、当該処理を中断処理保持部５ｄの待ち行列から取り除き、処理実行部５ｃに渡す。

なお、同一の実行優先度が与えられている処理が複数ある場合には、先に実行要求のあった処理から順に実行権が与えられる。

【００３５】

処理実行部５ｃは、実行処理決定部５ｂから渡された処理を実行し、処理要求受け付け部１によって割り込まれるまで、若しくは実行している処理が終了するまで継続して実行する。なお、処理が正常に終了した場合には、終了処理部１０に処理が正常終了したことを通知する。

【００３６】

中断処理保持部５ｄは、処理実行部５ｃで実行がされていたが、新たな処理要求を行った実行優先度の高い処理に実行権を奪われた処理、或いは優先度逆転対応部７から渡された処理を中断された状態で格納する待ち行列であり、当該待ち行列は実行優先度の高い処理から順に並べて格納した状態となっている。

なお、中断された状態とは、例えば、処理がデータに対してロックを確保している場合は、ロックを確保したままの状態を表している。

【００３７】

優先度逆転検出部６は、処理実行部５ｃが実行している処理、或いは、中断処理保持部５ｄが保持している処理といった実行途中の処理において優先度逆転が発生しているか否かを定期的にチェックし、優先度逆転が発生していた場合には、優先度逆転が発生している２つの処理を優先度逆転対応部７に渡す。

【００３８】

なお、優先度逆転の発生は、例えば、或る一定単位量のデータ毎に当該データに対してロックを確保している処理を把握するようにしておき、或る処理がデータを使用するときに当該データに対して他の処理がロックを確保しているかを否かによって検出できる。すなわち、本実施例では処理実行部５ｃが一つであり、従優先度占有方式によって処理を実行しているので、処理実行部５ｃが実行している処理の実行優先度が最も高く、当該処理が使用したいデータに対して他の処理がロックを確保している場合は優先度逆転が発生していることを表しており、そのため、処理が使用したいデータに対して他の処理がロックを確保しているか否かを検出することによって優先度逆転の発生が検出できる。

【００３９】

優先度逆転対応部７は、優先度逆転が発生した２つの処理間の実行優先度の差を算出する優先度差算出部と、実行優先度差と予め設定した規定値とを比較する優先度差判定部と、実行優先度差が規定値未満と判定されたことに基づいて対応する実行制御対応部と、実行優先度差が規定値以上と判定されたことに基づいて対応するアボート対応部とを有し、優先度逆転が発生した場合に、優先度逆転が発生している２つの処理間の実行優先度の差と予め設定された規定値と比較し、比較に基づいてＷＰ方式或いはＨＰ方式を選択して適用する。

【００４０】

すなわち、優先度算出部が優先度逆転検出部６から渡された高実行優先度の処理T_{high}と低実行優先度の処理T_{low}との実行優先度の差を算出し、優先度差判定部が算出した実行優先度差と規定値とを比較する。

比較した結果、実行優先度差が規定値未満の場合には、実行制御対応部が処理T_{low}に処理T_{high}と同じ実行優先度を設定し、処理T_{high}を中断処理保持部５ｄに渡す。次いで、実行制御対応部が実行処理決定部５ｂに実行権を与える処理を決定させる指示を出して処理実行部５ｃに処理T_{low}を実行させる。

一方、比較した結果、実行優先度差が規定値以上の場合には、アボート対応部が処理T_{low}をアボートして、実行待ち処理保持部５ａに渡す。この結果、処理T_{high}がデータへのロックを確保でき、実行が進行されることとなる。

【００４１】

ここで、処理T_{high}と処理T_{low}の実行優先度は処理情報管理テーブル３から参照される。

なお、上記した予め設定する規定値とは、２つの処理に優先度逆転が発生した場合に対応として、低実行優先度の処理をアボートするよりも低実行優先度の処理を先に実行させる方がいいといった臨界点での実行優先度差となっており、ユーザ或いはシステムが決定することもできる。

【００４２】

オーバーロード検出部８は、処理情報管理テーブル３から処理実行部５ｃが実行中の処理に対する期限Ｄを取り出し、時刻管理部（図示せず）から現時点の時刻ｔを受け取り、取り出した期限Ｄと受け取った時刻ｔとを比較する。その結果、期限Ｄと時刻ｔが同じか或いは期限Ｄより時刻ｔが遅い場合には、処理実行部５ｃが実行している処理をアボートしてオーバーロード対応部９へ渡す一方、期限Ｄの方が時刻ｔより遅い場合には処理実行部５ｃによる処理の実行を継続させる。

【００４３】

オーバーロード対応部９は、オーバーロード検出部８から処理を受け取った場合には、当該処理の実行を中止し、終了処理部１０に当該処理の実行が中止となったことを通知する。

【００４４】

終了処理部１０は、ＯＳ５から処理の正常終了の通知若しくはオーバーロード対応部９から処理の実行を中止する通知を受け取った場合には、受け取った通知をアプリケーション又はユーザに連絡する。

【００４５】

次に、本発明の第１実施例に係る処理管理システムの動作を説明する。

或る処理を実行させる要求が発せられると、処理要求受け付け部１が当該処理を受け取って格納し、順次格納している処理を優先度決定部２に渡す。

そして、優先度決定部２は処理要求受け付け部１から渡された処理の実行優先度を決定し、当該処理とともに決定した実行優先度を処理情報管理部４へ渡す。

【００４６】

処理情報管理部４は受け取った処理に関するテーブルを処理情報管理テーブル３に追加し、当該処理を実行待ち処理保持部５ａに渡す。実行待ち処理保持部５ａは受け取った処理を実行処理決定部５ｂによって実行権が与えられるまで格納する。

【００４７】

前記実行権は実行処理決定部５ｂが図３及び図４に示す動作を行うことによって与えられる。

まず、実行処理決定部５ｂは処理実行部５ｃに実行中の処理があるか否かを検出する（ステップＳ１）。

【００４８】

処理実行部５ｃに実行中の処理がない場合には、実行待ち処理保持部５ａに処理があるか否か（ステップＳ２）、更に、中断処理保持部５ｄに処理があるか否かを検出する（ステップＳ３、Ｓ４）。

実行待ち処理保持部５ａ、中断処理保持部５ｄの両方に処理が格納されていると検出した場合には、実行待ち処理保持部５ａに格納されている中で最も実行優先度の高い処理t_{wait}と、中断処理保持部５ｄに格納されている中で最も実行優先度の高い処理t_{int}との優先度を比較し（ステップＳ５）、実行優先度の高い方の処理に実行権を与える（ステップＳ６、Ｓ７）。

【００４９】

一方、実行待ち処理保持部５ａ、中断処理保持部５ｄのいずれか一方に処理が格納されていると検出した場合には、実行待ち処理保持部５ａ或いは中断処理保持部５ｄに格納されている処理の中で最も実行優先度の高い処理に実行権を与える（ステップＳ６、Ｓ７）。

なお、実行待ち処理保持部５ａ、中断処理保持部５ｄの両方に処理が格納されていないと検出した場合には、実行する処理が存在しないので実行権を与えない。

【００５０】

一方、処理実行部５ｃに実行中の処理がある場合には、実行待ち処理保持部５ａに処理が格納されているか否かを検出する（ステップＳ８）。
実行待ち処理保持部５ａに処理が格納されていると検出した場合には、実行中の処理t_{run}の実行優先度と、実行待ち処理保持部５ａに格納された処理の中で最も実行優先度の高い処理t_{wait}の実行優先度とを比較し（ステップＳ９）、実行待ち処理保持部５ａに格納された処理t_{wait}の実行優先度が実行中の処理t_{run}の実行優先度より高いときには、実行中の処理t_{run}を中断処理保持部５ｄに格納し（ステップＳ１０）、実行待ち処理保持部５ａに格納された処理t_{wait}に実行権を与える（ステップＳ１１）。
一方、実行待ち処理保持部５ａに処理が格納されていないと検出した場合或いは、処理が格納されていると検出して、実行待ち処理保持部５ａに格納された処理t_{wait}より実行中の処理t_{run}の実行優先度が高いと検出した場合には、実行中の処理t_{run}に実行権を継続して与える（ステップＳ１２）。

【００５１】
処理に対して実行権が与えられると、処理実行部５ｃによって前記処理が実行される。なお、処理が正常に終了した場合には終了処理部１０に処理が正常終了したことを通知する。

【００５２】
上記した処理実行部５ｃによって処理が実行されている際には、オーバーロード検出部８はオーバーロード状態が発生していることを検出し、優先度逆転検出部６は実行途中の処理に優先度逆転が発生していることを検出する。
すなわち、オーバーロード検出部８は、時刻管理部から現時点の時刻を受け取るとともに、処理管理テーブル３に管理されている実行中の処理に関する期限を受け取って、現時点の時刻が期限を超えてオーバーロード状態となっている場合には、実行中の処理をオーバーロード対応部９に渡す。

【００５３】
一方、優先度逆転検出部６は、処理実行部５ｃが実行している処理が中断処理保持部５ｄに保持されている処理によってデータへのロック確保をブロックされているか否かを検出して優先度逆転の発生を検出する。そして、優先度逆転が発生したことを検出した場合には、優先度逆転が発生している２つの処理を優先度逆転対応部７に渡す。

【００５４】
上記した実行処理部５ｃによる処理の実行中にオーバーロード検出部８によって処理がオーバーロード状態にあると検出され、当該処理をオーバーロード対応部７が受け取ると、オーバーロード対応部９は当該処理をアボートし、当該処理を実行しないことを終了処理部１０へ通知する。

【００５５】
一方、上記した実行処理部５ｃによる処理の実行中に優先度逆転検出部６によって優先度逆転が発生したことを検出され、優先度逆転が発生している２つの処理を優先度逆転対応部７が受け取ると、優先度逆転対応部７は以下に示す優先度逆転対応処理を行う。
まず、優先度逆転対応部７は受け取った２つの処理の実行優先度を処理情報管理テーブル３から読み込み（ステップＳ１３）、比較的高い実行優先度が指定された処理T_{high}と比較的低い実行優先度が指定された処理T_{low}との実行優先度の差ｄを算出する（ステップＳ１４）。

【００５６】
次いで、算出した実行優先度差ｄと予め設定された規定値とを比較し（ステップＳ１５）、実行優先度差ｄが規定値未満である場合には、処理T_{high}と処理T_{low}との実行順序に対する制約が弱く、処理T_{low}をアボートするよりも実行順序を換えて処理T_{low}を先に実行させる方が良いことを表しているので、処理T_{low}に処理T_{high}の実行優先度と同じ実行優先度を設定し、処理T_{high}を中断処理保持部５ｄに渡し、実行処理決定部５ｂに実行権を与える処理を決定させる指示を行う（ステップＳ１６）。この結果、実行処理決定部５ｂによって実行権が処理T_{low}に与えられ、処理T_{low}が処理実行部５ｃによって実行さ

れることとなり、その後、引き続いて処理T_{high}が実行されることとなる。

【００５７】

一方、実行優先度差ｄが規定値より大きい場合には、処理T_{high}と処理T_{low}との実行順序に対する制約が強く、処理T_{high}を先に実行させることが必要であるので、処理T_{low}をアボートして、実行待ち処理保持部５ａに渡す（ステップＳ１７）。この結果、処理T_{high}がデータに対してロックを確保することができ、実行が進行する。なお、アボートされた処理T_{low}は後に改めて最初から実行されることとなる。

【００５８】

処理実行部５ｃから処理を正常に終了したとの通知を受け取った場合或いは、オーバーロード対応部９から処理の実行をしないとの通知を受け取った場合には、終了処理部１０は、ユーザ或いはアプリケーションへ受け取った通知に基づく情報を知らせる。

【００５９】

上記のように、優先度逆転が発生した場合に、実行時間が分からない処理に対しても優先度逆転が発生した処理間の実行優先度差によって、実行優先度に従った実行順序に対する制約が強い場合には実行優先度に従った実行順序で処理の実行を行い、実行優先度に従った実行順序に対する制約が弱い場合には実行順序を守ることよりも計算機資源の無駄を防ぐ、といった適切な対応を選択できる。

【００６０】

なお、上記実施例では処理実行部５ｃが従優先度占有方式に従って処理を実行していたが、これに限らず、処理実行部５ｃがラウンドロビン方式等といった非占有方式において実行優先度に従って処理を実行するようにしてもよく、また、上記した従優先度占有方式と非占有方式とを処理によって切換えて実行するようにしてもよく、要は実行優先度に従って処理を実行する方式であればよい。

【００６１】

なお、非占有方式に従って処理を実行する場合においては、優先度逆転検出部６を、例えば、或る一定単位量のデータ毎に当該データに対してロックを確保している処理を把握するようにしておき、或る処理がデータを使用するときに当該データに対して他の処理がロックを確保しているかを否か検出し、ロックが確保されている場合に、ロックを確保した他処理との実行優先度を比較させるように構成しておくことで優先度逆転の発生を検出でき、本発明の効果を得ることができる。

【００６２】

本発明の第２実施例に係る処理管理システムを図６を参照して説明する。なお、図１で示した第１実施例に係る処理管理システムと同一部分には同一符号を付して重複する説明を省略する。

この処理管理システムは、第１実施例で示した処理管理システムにおいて、優先度逆転が発生している２つの処理の処理方式に応じて優先度逆転への対応をするようにしたものであり、優先度逆転が発生している処理の処理方式を判定して対応する優先度逆転判断部１１、を新たに備えたものである。

【００６３】

なお、ＯＳ５は処理実行部５ｃを占有して実行される占有処理及び処理実行部５ｃを占有せずに実行される非占有処理の実行を管理しており、占有処理が存在しない場合にのみ非占有処理を実行し、非占有処理を実行している際に新たな占有処理の要求があった場合には、非占有処理を直ちに中断して新たな占有処理を実行する。

【００６４】

優先度逆転判断部１１は、優先度逆転が発生している処理の処理方式を判定する優先度逆転判定部と、優先度逆転が占有処理と非占有処理との間に発生している場合に対応を行う第２アボート対応部とを有し、占有処理に指定する高実行優先度の範囲と、非占有処理に指定する低実行優先度の範囲とを把握しており、優先度逆転検出部６から渡された２つの処理のそれぞれが前記範囲のいずれに属するかを判定し、判定した結果に応じて図７に示す対応を行う。

【００６５】
すなわち、優先度逆転検出部６から受け取った高実行優先度の処理T_{high}と低実行優先度の処理T_{low}とが上記した範囲のいずれに属するかを優先度逆転判定部が判定する。
この結果、処理T_{high}が占有処理に指定する高実行優先度範囲に属し、処理T_{low}が非占有処理に指定する低実行優先度範囲に属すると判定された場合には、第２アボート部が非占有処理である処理T_{low}が占有処理である処理T_{high}をブロックしていることとなるので、非占有処理である処理T_{low}をアボートする。
一方、処理T_{high}及び処理T_{low}がともに非占有処理に指定する低実行優先度範囲に属すると判定された場合には、２つの処理は非占有処理であるので何も行なわず、処理T_{high}及び処理T_{low}がともに占有処理に指定する高実行優先度範囲に属すると判定された場合には、２つの処理を優先度逆転対応部７へ渡す。
【００６６】
次に、この処理管理システムの動作を説明する。なお、第１実施例と同様な動作の説明を省略する。
ＯＳ５が処理を実行している際に、優先度逆転検出部６が優先度逆転の発生を検出すると、優先度逆転の発生している２つの処理を優先度逆転判断部１１へ渡し、２つの処理を受け取ると優先度逆転判断部１１は、図８に示す優先度逆転への対応処理を行う。
まず、優先度逆転判断部１１は、受け取った処理の中で比較的高い実行優先度の処理T_{high}が占有処理であるか否かを検出し（ステップＳ１８）、さらに受け取った処理の中で比較的低い実行優先度の処理T_{low}が占有処理であるか否かを検出する（ステップＳ１９，Ｓ２０）。
【００６７】
処理T_{high}及び処理T_{low}がともに占有処理である場合には、２つの処理を優先度逆転対応部７へ渡す（ステップＳ２１）。
一方、処理T_{high}が占有処理であり、処理T_{low}が非占有処理である場合には、処理T_{low}をアボートして占有処理である処理T_{high}を実行可能にする（ステップＳ２２）。
なお、処理T_{high}及び処理T_{low}がともに非占有処理である場合には、何も行わない。また、処理T_{high}が非占有処理であり、処理T_{low}が占有処理である場合には、起こりえない場合であるのでエラー処理を行う。
【００６８】
次に、第２実施例に係る処理管理システムにおいて、ＯＳ５として市販のＯＳであるSolaris 2.5(Solarisは登録商標）を使用した例を第１実施例に係る処理管理システムと比較して説明する。なお、Solaris 2.5は実行優先度として０～１５９を有し、０～５９は非占有処理のための実行優先度、６０～９９は処理に対して指定することができないシステムのための実行優先度、１００～１５９は占有処理のための実行優先度となっている。
【００６９】
ここで、実行優先度”５９”が指定された非占有処理と、実行優先度”１００”が指定された占有処理との間に優先度逆転が生じた場合を例に取って説明する。
この場合、第１実施例に係る処理管理システムにおいては、例えば、規定値の取り方によって非占有処理が占有処理より先に実行される可能性が生じてくる。すなわち、規定値を”５０”と設定していると、優先度逆転が生じている上記２つの処理間の実行優先度の差が規定値”５０”未満なので、非占有処理が占有処理と同じ実行優先度に引き上げられ、非占有処理が占有処理より先に実行される。
このことは、終了期限等が与えられていたり、早期実行が望まれている占有処理に取っては問題である。
これに対して、第２実施例の処理管理システムにおいては、上記した優先度逆転が発生した場合には、必ず非占有処理をアボートするので、占有処理を直ちに実行することができる。
【００７０】
上記のように、優先度逆転が発生した場合に、占有処理を非占有処理に妨害されることな

く実行することができる。さらに、優先度逆転が発生した２つの占有処理に対して実行優先度差に基づいて、実行優先度に従った実行順序に対する制約が強い場合には実行優先度に従った実行順序で処理の実行を行い、実行優先度に従った実行順序に対する制約が弱い場合には実行順序を守ることよりも計算機資源の無駄を防ぐ、といった適切な対応を選択できる。

【００７１】

なお、上記第２実施例において優先度逆転対応部７を備えずとも、優先度逆転が発生した場合に常に占有処理を非占有処理に妨害されることなく実行することができる。

なお、上記した第１実施例及び第２実施例のＯＳ５は、Solaris 2.5、Windows NT 3.5(Windowsは登録商標）等といったＯＳを使用してもよい。

【００７２】

また、優先度逆転対応部７は、優先度逆転の対応として、低実行優先度の処理を高実行優先度の処理と同じ実行優先度に引き上げて設定していたが、高実行優先度の処理の実行優先度より高く設定するようにしてもよい。

また、オーバーロード対応部９は、受け取った処理の再実行を中止させる制御を行っていたが、例えば、受け取った処理の優先度を下げて再実行させる等といった他処理から計算機資源を無意味に奪ってしまうことのない制御を行ってもよい。

また、オーバーロード検出部８及びオーバーロード対応部９は備えずともよいが、備えることによって終了期限を超えて価値が著しく低下した処理が計算機資源を使用するといった他の処理の実行を妨げる状況を防ぐことができる。

【００７３】

なお、上記した各機能部１～１１は、ＣＰＵがプログラムに従って動作することにより構成されるが、これら機能部を独立したハードウエア回路として構成してもよい。

なお、上記した第１実施例及び第２実施例では処理実行部５ｃが一つであるシングルプロセッサについて説明したが、処理実行部５ｃを複数備えたマルチプロセッサに対しても本発明を適用できる。

【００７４】

【発明の効果】

以上説明したように本発明では、優先度逆転が発生した場合に、優先度逆転が生じている２つの処理間の実行優先度の差と予め設定した規定値との比較をして、ＨＰ方式或いはＷＰ方式を選択するようにしたために、実行優先度に忠実に処理を実行できるとともに、計算機資源の無駄を省くことができ、効率よく処理を実行ができる。

また、本発明では、優先度逆転が発生した２つの処理の種類を検出し、２つの処理が占有処理、非占有処理の場合に非占有処理をアボートするようにしたために、占有処理を常に非占有処理より先に実行することができ、占有処理に適した実行ができる。

【図面の簡単な説明】

【図１】　第１実施例に係る処理管理システムの構成図である。

【図２】　第１実施例に係る処理情報管理テーブルを説明する図である。

【図３】　第１実施例に係る実行処理決定部が行う動作を説明する第１フローチャートである。

【図４】　第１実施例に係る実行処理決定部が行う動作を説明する第２フローチャートである。

【図５】　第１実施例に係る優先度逆転への対応動作を説明するフローチャートである。

【図６】　第２実施例に係る処理管理システムの構成図である。

【図７】　第２実施例に係る優先度逆転への対応動作を説明する図である。

【図８】　第２実施例に係る優先度逆転への対応動作を説明するフローチャートである。

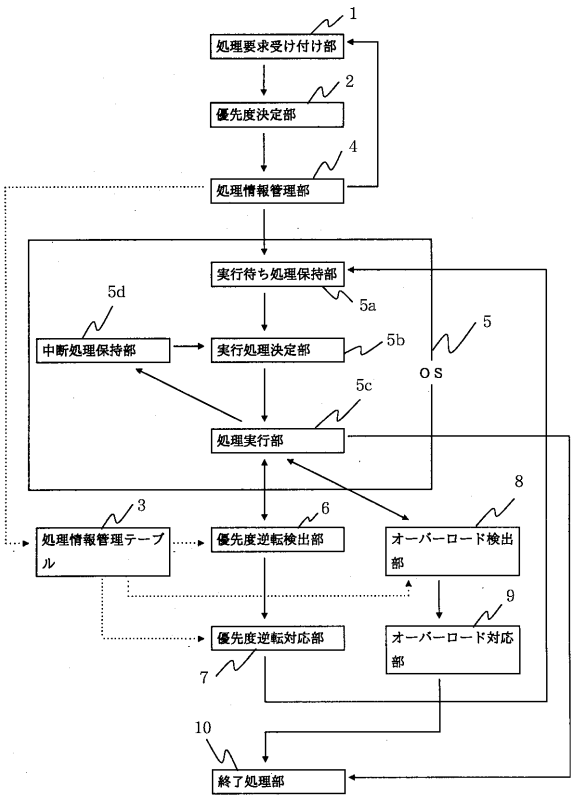
【符号の説明】

５・・・ＯＳ、　　５ｃ・・処理実行部、

６・・優先度逆転検出部、　　７・・優先度逆転対応部、

１１・・優先度逆転判断部、

【図１】



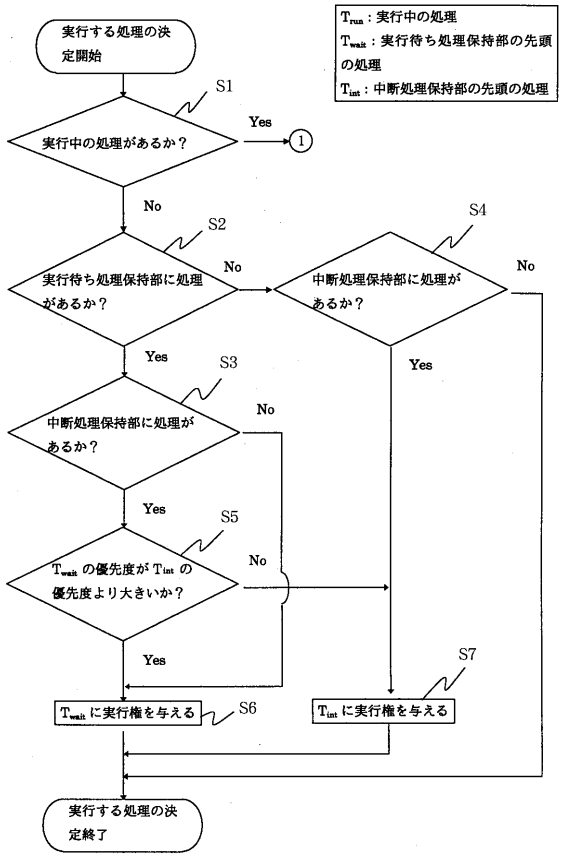
【図２】

処理識別子	期限	優先度
２３６５	１５：５５	７
２３７０	９：２８	２２
.....		

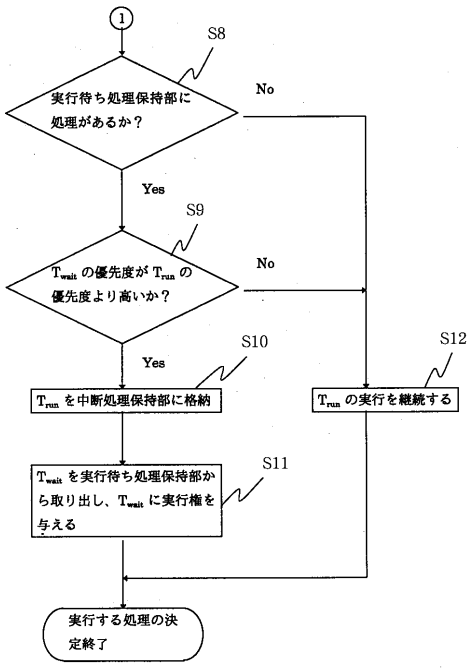
【図７】

		Thigh	
		L	H
Tlow	L	何もしない	HP方式
	H		第１実施例の発明の方式

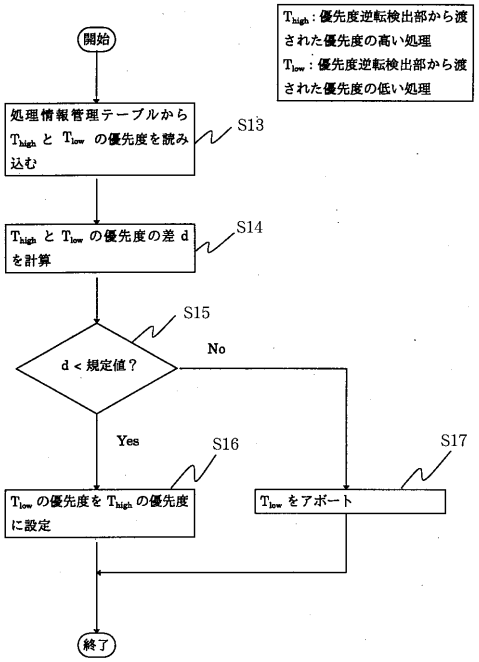
【図３】



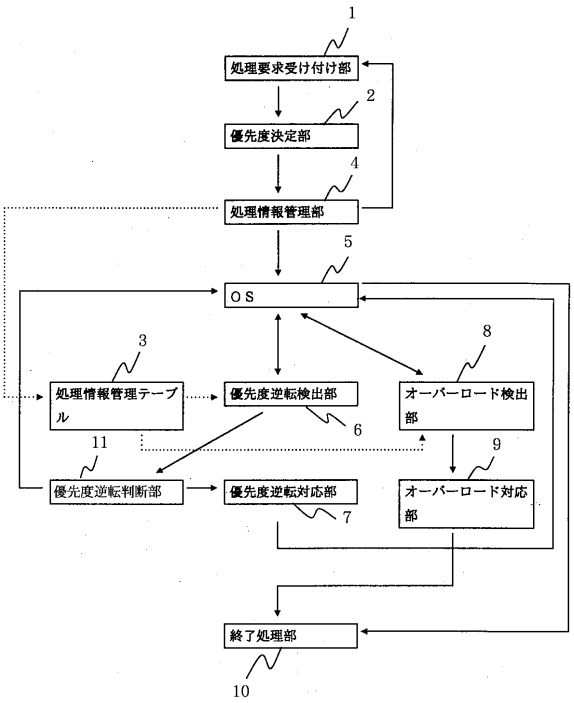
【図４】



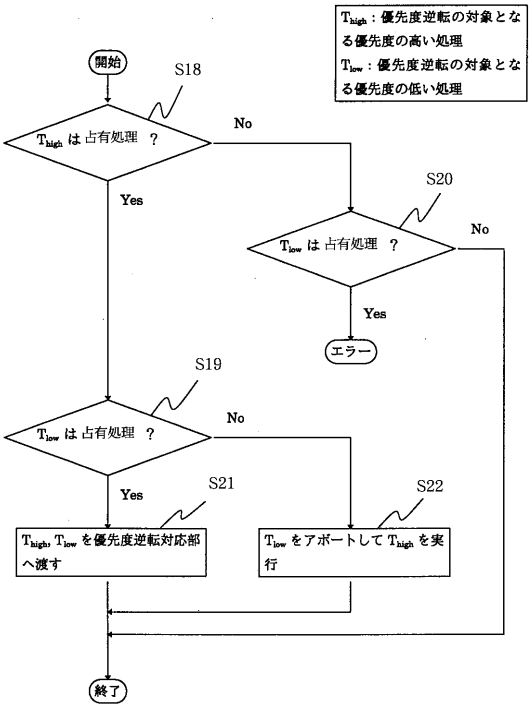
【図５】


T_{high}：優先度逆転検出部から渡された優先度の高い処理
T_{low}：優先度逆転検出部から渡された優先度の低い処理
開始
処理情報管理テーブルから T_{high} と T_{low} の優先度を読み込む
S13
T_{high} と T_{low} の優先度の差 d を計算
S14
S15
d < 規定値？
No
Yes
S16
S17
T_{low} の優先度を T_{high} の優先度に設定
T_{low} をアボート
終了

【図６】


1
処理要求受け付け部
2
優先度決定部
4
処理情報管理部
5
OS
3
処理情報管理テーブル
優先度逆転検出部
6
8
オーバーロード検出部
11
9
優先度逆転判断部
優先度逆転対応部
7
オーバーロード対応部
終了処理部
10

【図８】


T_{high}：優先度逆転の対象となる優先度の高い処理
T_{low}：優先度逆転の対象となる優先度の低い処理
開始
S18
T_{high} は 占有処理 ？
No
Yes
S20
T_{low} は 占有処理 ？
No
Yes
エラー
S19
T_{low} は 占有処理 ？
No
Yes
S21
S22
T_{high}, T_{low} を優先度逆転対応部へ渡す
T_{low} をアボートして T_{high} を実行
終了

フロントページの続き

(56)参考文献 特開平０８－０７７０２５（ＪＰ，Ａ）
特開平０２－２０４８３８（ＪＰ，Ａ）
特開昭６２－２７１１４７（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06F 9/46-9/48