

B-38

ITRON 仕様カーネルのレディキューに関する性能比較
Performance Comparison of Ready Queue of ITRON Method Kernel

渡辺 洋輔† 伊藤 圭† 宮内 新† 荒井 秀一†
Yosuke Watanabe Kei Ito Arata Miyauchi Shuichi Arai

1. はじめに

OS の主要な機能の一つに、次に実行されるタスクを決定するスケジューリングがある。μITRON4.0 仕様のスケジューリング規則では、優先順位の高いタスクが実行できる状態にある限り、それより優先順位の低いタスクは全く実行されない。実装はレディキューによって実現する。

レディキューの内部構造に関して仕様では定めていないので、実装する際に決定することになる。その際のガイドラインがあることが望ましい。そこで、レディキューの構造による性能の差を定量的に評価することを本研究の目的とする。

2. 比較するレディキュー構造

2.1 優先度ごとのレディキュー

現在、豊橋技術科学大学を中心に開発が進められている TOPPERS/JSP カーネルでは、優先度段数に応じたレディキューを有し、各優先度ごとに双方向のリンクリストを形成している。各優先度のタスクの存在状況を示すビットマップを用意し、タスクにアクセスする際にはビットマップを参照する。

2.2 シングルレディキュー

この構造に対して簡略化した構造として、全ての実行可能状態のタスクを一本のリンクリストにしたレディキュー構造「シングルレディキュー」構造を作成した。レディキューが優先度ごとに存在していたために必要とされていたビットマップ及びビットマップサーチ関数、最高優先順位タスク探索関数などは不要となるので削除し、タスク挿入場所を線形探索する探索関数などを作成し付加した。レディキューの操作に関連する関数も構造変化に伴い改変を行った。

3. 実現環境

性能測定に使用した実機は日立超 LSI システムズの Solution Engine(SH3 133MHz で動作)を使用した。GNU 開発環境をもちいて PC 上でクロスコンパイルをおこない、実機と PC を通信させて測定をおこなった。

4. 性能測定結果

タスク操作関連のオブジェクトサイズとシステムコール発行の際のオーバーヘッドを性能評価の指標とした。

4.1 オブジェクトサイズ

タスク管理関連のオブジェクトモジュールのサイズはシングルレディキューを実装したことにより、24.8kbyte から 23.0kbyte に減少した。

4.2 オーバーヘッドの計測

また測定するオーバーヘッドを平均時間と最悪時間の 2 種類で比較した。試行回数はシステムコールとレディキュー状態の組み合わせに対し各 1000 回とした。オーバーヘッド

† 武蔵工業大学 Musashi Institute of Technology

はシステム評価用システムコールを用いて測定した。オーバーヘッドを測定するシステムコールはタスク起動命令、優先度変更命令、レディキュー回転命令の 3 種類において測定をおこなった。

4.2.1 タスク起動命令

キャッシュを使用時の結果を図 1 に示す。

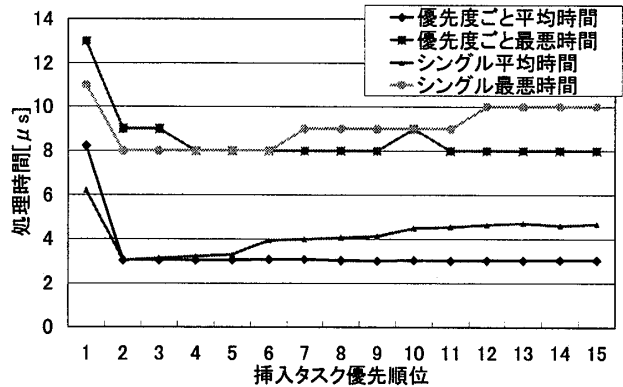


図 1 タスク起動命令 (キャッシュオン)

起動したタスクがレディキューに挿入される。挿入されたタスクが最高優先順位タスクならばディスパッチが起こるので処理時間が增大する。最高優先順位タスクの場合にはシングルレディキュー構造のほうが平均時間、および最悪時間でも優先度ごとのレディキュー構造を上回る。この結果からはシングルレディキュー構造は平均時間では優先度ごとのレディキュー構造と同等、あるいは下回っている。しかし、最悪時間の面から見ると、優先度ごとのレディキュー構造でも

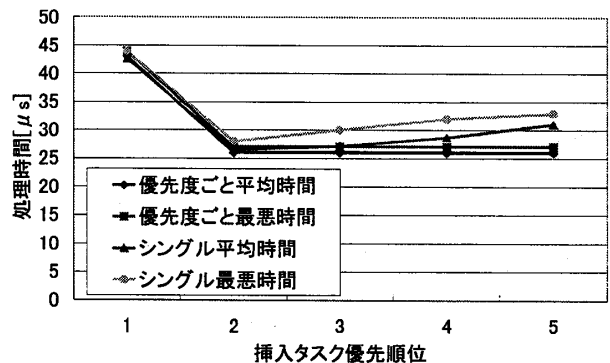


図 2 タスク起動命令 (キャッシュオフ)

9 μs の処理時間がかかっている場合もある。最悪時間の面からはタスク数が 10 個まで双方のレディキュー構造は同等の性能を持つと考えられる。タスク数が 10 を超えてくると優先度ごとのレディキュー構造のほうが平均時間、最悪時間ともにシングルレディキュー構造を上回る。

キャッシュをオフにした場合の結果を図2に示す。最高優先順位タスクの起動には2つの構造の間で性能差は見られず、優先順位が下がるにつれて優先度ごとの構造が優位になった。

4.2.2 タスク優先度変更命令

対象タスクが最高優先順位のタスクでない場合に最高になるように優先度を変更、もしくは最高優先順位のタスクの優先度を他のタスクの優先度以下に下げた場合、ディスパッチが起こる。図3と図4に示す結果にはこの値を用いた。

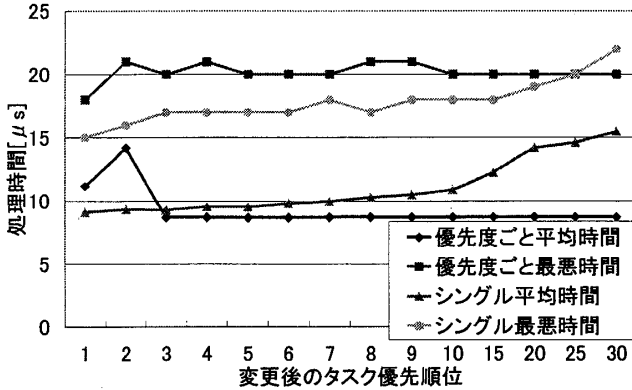


図3タスク優先度変更命令 (キャッシュオン)

変更後の優先順位が最高優先順位に極めて近い場合にはシングルレディキュー構造は優先度ごとのレディキュー構造を平均時間、最悪時間ともに上回った。そうでない場合には平均時間では優先度ごとのレディキュー構造が上回っているが、最悪時間の比較においては優先順位が25程度までシングルレディキュー構造の方が上回るといった結果になった。

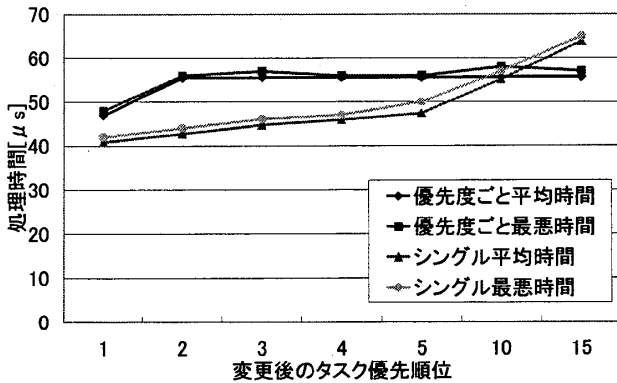


図4タスク優先度変更命令 (キャッシュオフ)

キャッシュをオフにした場合、シングルレディキューの性能が上回るタスク数の限界がタスク数10程度までになっているが、平均時間においてもタスク数10程度までと逆に多くなっている。

4.2.3 レディキュー-回転命令

レディキュー回転命令は指定した優先度のタスク群を回転させる。最高優先度のタスク群を回転させたときの結果を図5と図6に示す。

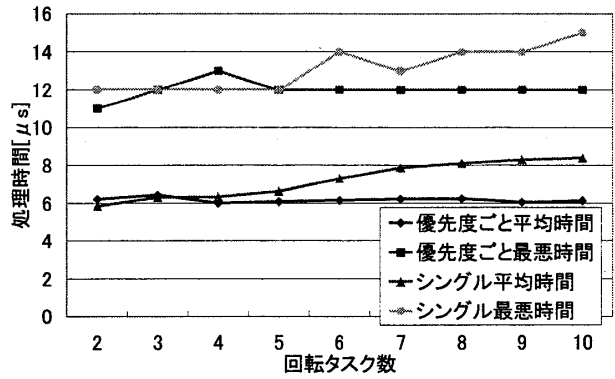


図5レディキュー回転命令 (キャッシュオン)

結果から2つの構造の性能が同程度といえるのは回転させるタスク数がせいぜい5個程度までであった。それ以上になると優先度ごとのレディキュー構造の方が平均時間、最悪時間ともに上回る。また、この結果は回転させる優先度より低い優先度のタスク数には依存しなかった。

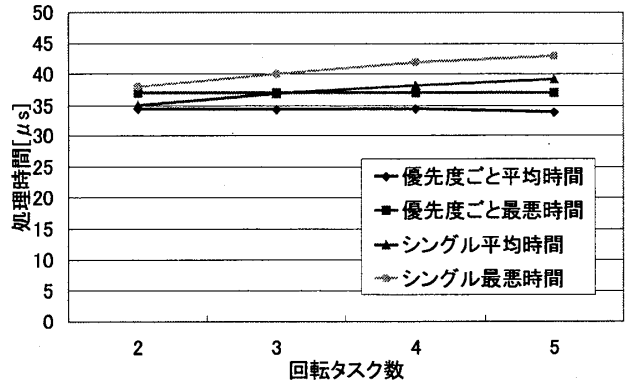


図6レディキュー回転命令 (キャッシュオフ)

キャッシュをオフにした場合にはタスク数が少ない場合においても優先度ごとのレディキュー構造の性能が上回った。

5. むすび

作成、実装したシングルレディキューは優先度キュー構造で不必要となった機能を排除することによってオブジェクトサイズの減少を図ることができた。シングルキューは優先度キューと比較してタスク数が少ないほど有利である。測定の結果今回対象とした3つのシステムコールからはタスク数5個までならばシングルレディキュー構造の性能が上回ってくるという実装上の指針が得られた。また、主にキャッシュの使用時においてシングルレディキューの有効性が見られた。

参考文献

[1] (社) トロン協会 ITRON 部会 坂村 健 高田 広章: μITRON4.0 仕様書
 [2] 坂村 健: 「ITRON 標準ガイドブック 2」 1994
 [3] 株式会社 日立製作所: SH7709A ハードウェアマニュアル 1999