

μ ITRON4.0 仕様 OS における ミューテックスの実装

2 L-2

樋口 正雄* 高田 広章† 石川 知雄*

* 武蔵工業大学大学院工学研究科 † 豊橋技術科学大学情報工学系

1 はじめに

わが国における機器組込み用リアルタイムオペレーティングシステム (RTOS) のデファクトスタンダードとなっている ITRON の最新の仕様である μ ITRON4.0 仕様 (1999 年) には, ハードリアルタイムシステムへの適応を考慮した規定が加えられている [1]. この μ ITRON4.0 仕様の評価に加え, 研究・教育用プラットフォームの提供を目的とした “TOPPERS/JSP カーネル” が豊橋技術科学大学組込みリアルタイム研究室を中心とした TOPPERS プロジェクトにより開発された.

今回, μ ITRON4.0 仕様で新たに規定が加わった拡張同期・通信機能である ミューテックスについて定義した仕様の評価を目的として, TOPPERS/JSP カーネル上に実装したので報告する.

2 μ ITRON4.0 の ミューテックス

応答時間に関する厳しい制約を課されるハードリアルタイムシステムの構築において, 排他制御に伴う「上限のない優先度逆転現象」がしばしば問題となる. ミューテックスは優先度逆転現象の防止を目的として, 優先度継承プロトコルと優先度上限プロトコル [2] をサポートする排他制御機構である. 具体的には, 資源をロックするタスクの優先度を一時的に上昇させることにより同プロトコルを実現する.

μ ITRON4.0 では, 優先度継承・上限プロトコルの定義に従った「厳密な優先度制御規則」に加え, 上昇させた優先度をもとに戻すタイミングを限定した「簡略化した優先度制御規則」を規定し, どちらを採用するかは実装依存としている. この 2 種類の制御規則のミューテックスを実装し性能を比較した上, 仕様の有効性を検証することを目的としている.

3 実装

3.1 データ構造

新たに TOPPERS/JSP カーネル上に実装したミューテックス管理ブロック (MTXCB) の構造を示す.

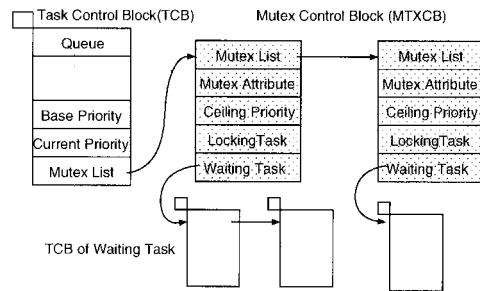


図 1: ミューテックス管理ブロックとタスク管理ブロック

ミューテックス管理ブロック セマフォと大きく異なる点は, ロックを獲得しているタスクをミューテックス側から管理する必要があるということである. ミューテックス属性に TA_INHERIT を指定すると優先度継承プロトコルを, TA_CEILING を指定すると優先度上限プロトコルをサポートする.

タスク管理ブロック内のフィールド タスク優先度をベース優先度 (ユーザーアプリケーションで与えられる優先度) と現在優先度 (優先度継承で変化する実効的な優先度) とに区別して管理する. また, タスク側からもロックしているミューテックスを管理する必要があるため, ミューテックスリストを追加した.

3.2 優先度制御

優先度を上げるタイミング

- TA_CEILING 属性ミューテックスのロック獲得
- TA_INHERIT 属性ミューテックスをロックしているタスクより高い優先度のタスクのロック待ち状態への移行
- TA_INHERIT 属性ミューテックス待ちのタスクの優先度変更 (上昇)

Implementations of Mutex on μ ITRON4.0-specification Kernel

Masao HIGUCHI*, Hiroaki TAKADA†, Tomo ISHIKAWA*

*Musashi Institute of Technology

†Toyohashi University of Technology

優先度を下げさせるタイミング

- ミューテックスの開放
- タスクのミューテックス待ち状態の強制解除
- TA_INHERIT 属性ミューテックス待ちのタスクの優先度変更 (降下)

優先度を下げる処理においては、ロックしているミューテックスとその待ちタスクを探索し、新しい優先度値を求める操作が必要となる。しかし、簡略化した優先度制御規則では、優先度を下げるタイミングを全てのミューテックスを開放した時に限定しているため、探索に要していたオーバーヘッドの減少が見込まれる。

4 評価

日立 SH7709A (SH-3 クロック周波数 133MHz) 上で基本的なミューテックス処理サービスコールの応答時間を測定した。(キャッシュOFF)

1. 対象ミューテックスが他のタスクによってロックされていない場合の loc_mtx サービスコールの応答時間。
2. 同上。ただし、対象ミューテックスが TA_CEILING 属性であり、タスクの優先度を上限優先度に上げる操作も含む。
3. 対象ミューテックスが既に他のタスクによってロックされていて、待ち状態に入る場合の loc_mtx サービスコールの応答時間。タスクディスパッチ操作も含む。
4. 同上。ただし、対象ミューテックスが TA_INHERIT 属性であり、先にロックしているタスクが待ちタスクの優先度を継承する操作も含む。
5. 同上。対象ミューテックスを先にロックしているタスクが別のミューテックス待ち状態であった場合、4. の優先度継承に加え、そのミューテックスをロックしているタスクも優先度を継承する。(遷移的な優先度継承)
6. unl_mtx サービスコールの応答時間。ロック待ちタスクは存在しない。
7. unl_mtx サービスコールの応答時間。ロック待ちタスクの待ち解除操作を含むが、タスクディスパッチは生じない。
8. unl_mtx サービスコールの応答時間。ロック待ちタスクの待ち解除操作とタスクディスパッチも含む。
9. 同上。他にもロックしているミューテックスがある場合。

以下が、各テストプログラムにおけるサービスコールの応答時間である。対応するセマフォ処理サービスコールの応答時間も示す。

表 1: ミューテックス処理の応答時間 (単位: μ s)

	セマフォ	厳密な制御規則	簡略化制御規則
1	25	32	32
2	-	49	49
3	72	76	76
4	-	90	90
5	-	103	103
6	27	43	41
7	43	55	61
8	58	69	74
9	-	115	54

また、複数の TA_CEILING 属性ミューテックスをロックしている際の unl_mtx サービスコール処理時間の変化を図 2 に示す。

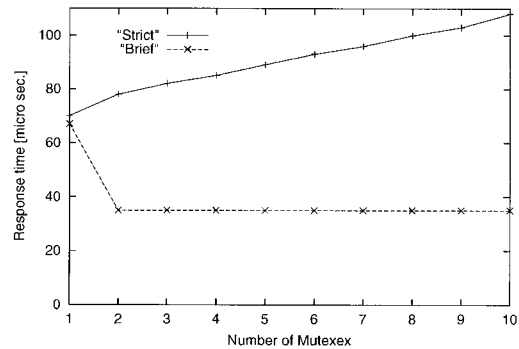


図 2: ロック中ミューテックス数と unl_mtx サービスコール処理時間

5 おわりに

μ ITRON4.0 仕様の評価を目的として TOPPERS/JSP カーネル上に 2 種類の優先度制御規則のミューテックスを実装し、基本的なシステムコールの応答時間による評価を行った。今後はアプリケーション・モデルを想定した評価を行い、最終的にはミューテックス実装法に関するガイドラインの提案を行いたいと考えている。

参考文献

- [1] 坂村健 監修/高田広章 編: “ μ ITRON4.0 仕様”, トロン協会 ITRON 部会 (1999)
- [2] L.sha, R.Rajkumar, and J.P.Leoczky: “Priority Inheritance Protocols: An Approach to Real-time Synchronization”, IEEE Trans. Computers, vol.39, pp.1774-1885, Sept. 1990.