

マルチプロセッサにおけるリアルタイムタスクの スケジューリングアルゴリズムの提案

西山 真穂 兪 明連 横山 孝典 志田 晃一郎

武蔵工業大学

1 はじめに

命令や要求に対し、定められた時間内に応答・処理を行うリアルタイムシステムというものがある。リアルタイムシステムにおいて、単一プロセッサによるタスクの最適なスケジューリング方法として EDF(Earliest Deadline First)アルゴリズム[1](以下 EDF)が確立されている。しかし、マルチプロセッサの環境下で EDF を用いたとき、タスクのデッドラインを守れない場合がある。また、マルチプロセッサの環境下ではほぼ最適な性能をもつスケジューリング方法として LLF(Least Laxity First)アルゴリズム[2](以下 LLF)があるが、等しい余裕時間を持つタスクが複数ある場合、コンテキストスイッチが頻発するという問題がある。

マルチプロセッサの環境下における上記の問題点を改善するため、余裕時間がゼロとなるタスクが発生するまで EDF を用い、余裕時間がゼロであるタスクがある間 LLF を用いる、ED/LL(Earliest Deadline/Least Laxity)アルゴリズム[3](以下 ED/LL)が提案されている。平均的に、LLF よりもコンテキストスイッチの発生回数は少ないが、場合によっては LLF よりも増えることがある。

そこで本研究の目的は、周期的タスクを対象にして LLF の問題点を改善したスケジューリングアルゴリズムを提案し、LLF よりもコンテキストスイッチの発生を抑えることである。

2 提案アルゴリズム

2.1 アルゴリズム概要

タスクの余裕時間に着目してスケジューリングを行っていく LLF を基にコンテキストスイッ

チの発生を抑えるため、アルゴリズムを以下のように取り決めた。

- 余裕時間が短いタスクを基本的に優先する。
- 余裕時間の同じタスクが複数ある場合は、デッドラインの近いタスクを優先する。
- 余裕時間が同じタスクが複数ある時刻に、余裕時間が同じであったが実行されなかったタスクの余裕時間がゼロになるまで、実行中のタスクを続行する。
- 余裕時間が同じタスクが複数ある時刻に、値が大きいため実行されなかったタスクは、時間の経過とともに余裕時間が短くなる。実行中のタスクと実行されなかったタスクの余裕時間が同じになったとき、再度優先度の割り当てを行う。
- 余裕時間がゼロであるタスクを最優先する。

以上のようなアルゴリズムに従ってスケジューリングを行うことで、コンテキストスイッチの発生回数を減らすとともに、各タスクのデッドラインを守ることが出来る。

2.2 スケジューリング例

マルチプロセッサ環境下において LLF を用いてスケジューリングを行った際、コンテキストスイッチの発生回数が特に増えやすいとされる、表 1 に示すようなタスクの余裕時間が全て同じ状態である周期的タスクのタスク集合を用いて、LLF と提案アルゴリズムによるスケジューリング例を以下に示す。また、それぞれでスケジューリングを行った結果より、コンテキストスイッチの発生回数の違いを示す。

タスク	実行時間	デッドライン	余裕時間
1	3	5	2
2	4	6	2
3	3	5	2

表 1 に示したタスク集合を提案アルゴリズムによってスケジューリングしている動きを、数値でトレースしたものを表 2 に示す。

Scheduling Algorithm for Real-time Task on Multiprocessor
Maho Nishiyama, Myungryun Yoo, Takanori Yokoyama,
Koichiro Shida,
Musashi Institute of Technology

表2 提案アルゴリズムによるタスク処理のトレース

時刻 t	(残り実行時間, デッドラインまでの時間, 余裕)			実行プロセッサ	
	タスク 1	タスク 2	タスク 3	P_1	P_2
0	(352)	(462)	(352)	1	3
1	(242)	(451)	(242)	1	3
2	(132)	(440)	(132)	1	2
3	(02-)	(330)	(121)	3	2
4	(01-)	(220)	(01-)	-	2
⋮	⋮	⋮	⋮	⋮	⋮

P_i : i 番目のプロセッサ j : i 番目のタスク t : 時刻

表 1 に示したタスク集合を, 時刻 $t = 6$ まで LLF によりスケジューリングを行った結果, 図 1 に示すようにコンテキストスイッチは 3 回発生した. 同様に, 提案アルゴリズムによりスケジューリングを行った結果, コンテキストスイッチの発生は 1 回で済み, LLF よりも減っている.

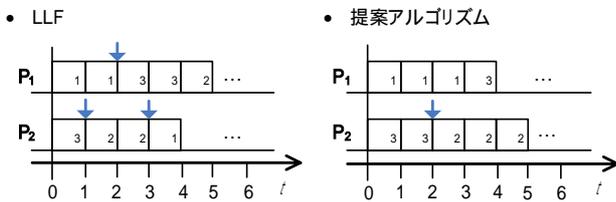


図1 LLF, 提案アルゴリズムによる処理の過程

3 実験・評価方法

3.1 実験方法

ランダムに実行時間, 周期の初期値を決めた周期的タスク集合を使用する. このタスク集合を用いて, シミュレーションによりスケジューリングを行う. このとき, 全タスクの一番初めの起動時刻は $t = 0$ とする.

デッドラインは周期と同値であるとしてタスク集合を作成する. 周期的タスク集合内のタスク数を決めるため, EDF のスケジューリング可能性の基準式を用いる. 式を以下に示す.

$$\sum_{i=1}^m \frac{C_i}{T_i} \leq 1, \quad i \text{ ————— (1)}$$

C_i : タスク i の実行時間

T_i : タスク i の周期

m : タスク数

与えられた周期的タスク集合が, 単一プロセッサの環境下において EDF によってスケジュール可能であるとき(1)式が成立する. (1)式内の右辺はプロセッサの使用率である. 右辺の値が 1, つまりプロセッサの使用率が 100% 以下であれば, EDF はデッドラインをオーバーしない. プロセッサ数を 2 つとするようなマルチプロセッサの環境である場合, この値を 2 とすればよい. このよ

うな(1)式を満たす m 個のタスクを持つタスク集合をスケジューリングに用いる.

今回(1)式を用いたのは, LLF は決まった基準式を持たないが, スケジューリングの可能性が(1)式にほぼ該当するためである.

3.2 評価方法

現時点では, 作成した周期的タスク集合を用いて EDF, LLF, ED/LL, 提案アルゴリズムにより数パターンのスケジューリングを行い, デッドラインを守れているかを検証し, 提案アルゴリズムがデッドラインを守っていることを確認した. EDF も検証の対象としたのは, (1)式の条件を満たしていてもマルチプロセッサの環境下ではデッドラインをオーバーする可能性があるからである.

次に, LLF, ED/LL, 提案アルゴリズムのコンテキストスイッチの発生回数を比較し, 提案アルゴリズムが LLF よりも発生回数を抑えたとともに, LLF よりも増えないことを確認した. また, ED/LL が LLF より多くコンテキストスイッチが発生したときでも, 提案アルゴリズムでは回数が減っていることを確認した.

4 おわりに

本研究では余裕時間が同じであるタスクが複数あった場合に, LLF ではコンテキストスイッチが増えるという問題を改善したスケジューリングアルゴリズムを提案し, そのプロトタイプを作成した.

今後, 提案アルゴリズムの評価を更に行っていく予定である. また, よりコンテキストスイッチの発生回数を抑えられるよう, アルゴリズムの追加を検討している.

参考文献

- [1] C.L.Liu, James W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment", *J.A.C.M.*, Vol.20, No.1, pp.46-67, 1973
- [2] Michal L.Dertouzos, Aloysius Ka-Lau Mok, "Multiprocessor On-Line-Scheduling of Hard-Real-Time Task", *IEEE. Trans. Software. Eng.*, Vol.15, No.12, pp.1497-1506, Dec, 1989
- [3] Seongje Cho, Suk-Kyoon Lee, Sang Ahn, Kwei-Jay Lin, "Efficient Real-Time Scheduling Algorithms for Multiprocessor Systems", *IEICE Trans. Commun.* Vol.E85-B, No.12, pp.2859-2867, Dec, 2002