

EDCL に様々な準最高優先度を設定するスケジューリングアルゴリズムの提案

辻井 智紀[†] 兪 明鍊[†]
Tomoki Tsujii Myungryun Yoo

1. 概要

近年の組込みリアルタイムシステムは、数多の技術発展により高性能なプラットフォームが必要になってきており^[1]、マルチプロセッサ環境で CPU を 100% 利用できる最適スケジューリングアルゴリズムが必要である。しかしながら、現在 CPU を 100% 利用できるアルゴリズムは、アルゴリズムが複雑であり、コンテキストスイッチなどによるオーバーヘッドが非常に大きいことから、実用的でないとされている。従来の研究で EDCL^[2] では EDF^[3] と比較してかなりデッドラインミス削減することができたが依然デッドラインミス量が多い。本論文では、最高優先度のほかに優先度を設定することにより、どれだけデッドラインミス削減できるかを実験した。その結果、プロセッサ数 4 の時は提案手法 2 がデッドラインミスの量をより削減することができ、コンテキストスイッチの削減は提案手法 4 が優れており、スケジューラの起動回数は提案手法 1 がほかのアルゴリズムよりも抑えることができた。

2. 従来手法

従来手法として EDF, EDCL を紹介する

2.1 EDF (Earliest Deadline First)

デッドラインが最も近いタスクに最高の優先度を与える動的優先度方式のスケジューリングアルゴリズムである。スケジューラの起動はジョブの起動時と終了時に行われる。EDF の動作例を図 2.1 に示す。実行するプロセッサ数は 2 であり、スケジューラの起動はタスクの開始時と終了時である

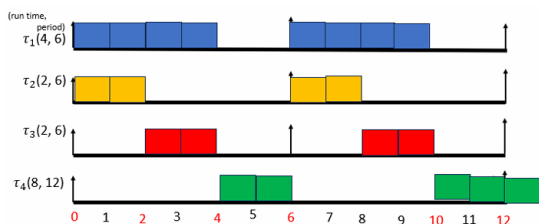


図 2.1. EDF の実行例

2.2 EDCL (Earliest Deadline Critical Laxity)

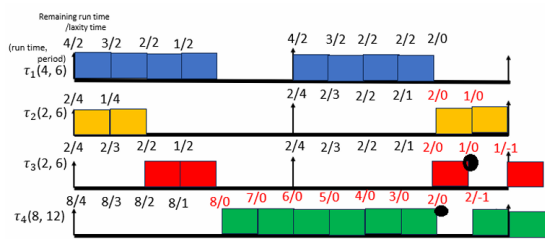


図 2.2. EDCL の実行例

EDCL は EDF と同様にスケジュールを行うが、余裕時間がクリティカルになったジョブに最高優先度を与えるアルゴリズムである。余裕時間がクリティカルとはある時刻 t で最もデッドラインが早いタスクの残り実行時間を求め、それよりも小さい余裕時間を持つタスクのことである。

3. 提案手法

3.1 提案手法 1

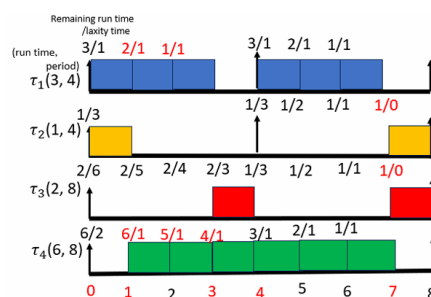


図 3.1. 提案手法 1

提案手法 1 は EDCL をベースにある時刻におけるデッドラインの最も早いジョブの中で残り実行時間が最大のジョブに準最高優先度を設定するアルゴリズムである。図 3.1 は実行例を示している。

3.2 提案手法 2

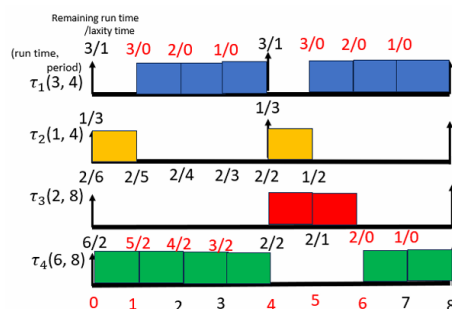


図 3.2. 提案手法 2

提案手法 2 は林の手法^[4]をベースに周期の長いタスクを次のスケジューラが起動するまで 1 度だけ準最高優先度を与え周期の長いタスクのデッドラインミスを削減し、のちに林の手法を同じ動作をする手法である。林の手法とは、EDCL をベースに、デッドラインミスが最も早いタスクのうち、最小残り実行時間を持つタスクに準最高優先度を与える手法である。

3.3 提案手法 3

提案手法 3 は、EDCL をベースに、タスク利用率が高いジョブに準最高優先度を与える手法である。図 3.3 にその手法の実行例を示す。

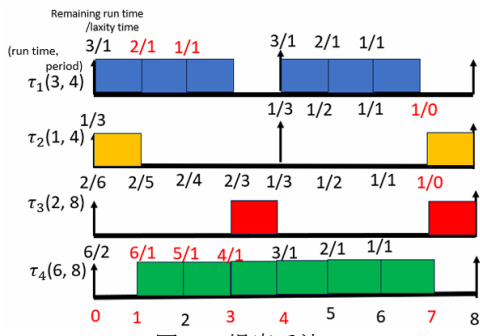


図 3.3. 提案手法 3

3.4 提案手法 4

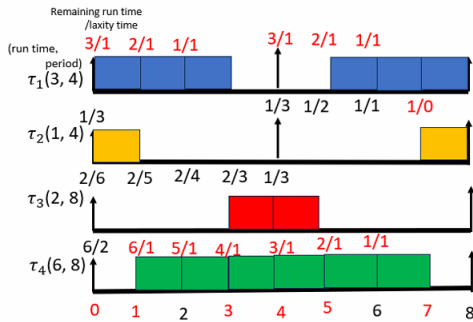


図 3.4. 提案手法 4

提案手法 4 は、EDCL をベースにある時刻におけるデッドラインの最も早いジョブの中で、周期の長いタスクに準最高優先度を与え実行を優先する。図 3.4 にこの手の実行例を示す。

4. 結果

シミュレーションによる評価と得られた結果の考察を述べる。デッドラインミス量、コンテキストスイッチ回数とスケジューラ起動回数を評価対象とし、対象アルゴリズムは EDCL、林の手法、提案手法 1 から 4 とする。プロセッサ数は 4 つとする。実行タスクセット数は 100 とする。

4.1 デッドラインミス量

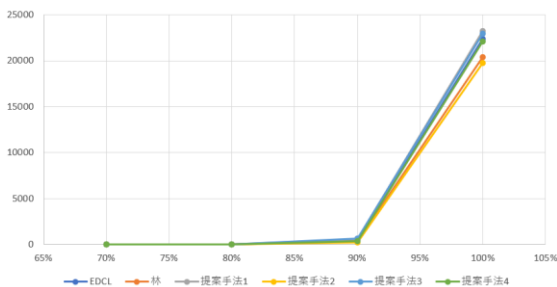


図 4.1. デッドラインミス量

図 4.1 は、横軸はシステム利用率、縦軸はデッドラインミス量を示している。提案手法 2 がすべての手法においてデッドラインミス量を削減することができた。実行されれば、周期の長いタスクに 1 つのプロセッサを占有することにより、タスク処理を分散することが可能となったため、デッドラインミスを最小限に抑えられたと考えている。

4.2 コンテキストスイッチ量

図 4.2 は、横軸はシステム利用率、縦軸はコンテキストスイッチ回数を示している。図 4.2 より、提案手法 4 が最もコンテキストスイッチを削減したことが分かる。

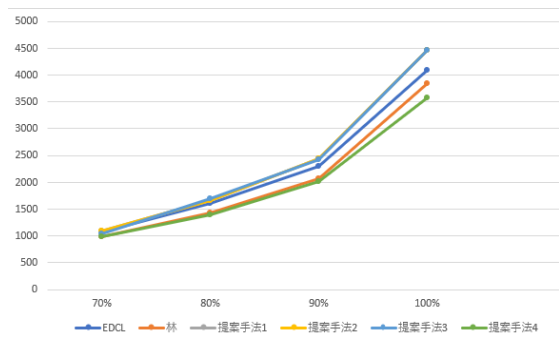


図 4.2. コンテキストスイッチ回数

提案手法 4 は周期が長いタスクに優先度を上昇させることによりコンテキストスイッチの回数を抑えられたと考えている。

4.3 スケジューラ起動回数

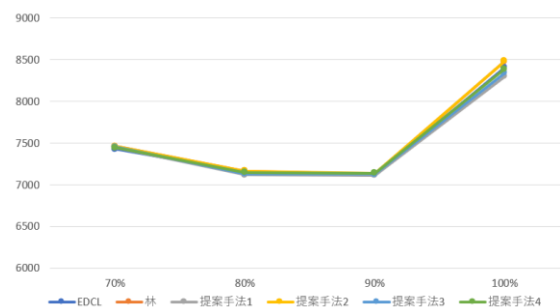


図 4.3. スケジューラ起動回数

図 4.3 は、横軸はシステム利用率、縦軸はスケジューラ起動回数を示している。これにより提案手法 1 がスケジューラ起動を抑えられたことが分かる。余裕時間が少ないタスクに準最高優先度を与えるとプロセッサを占有する時間が増えることによりスケジューラの起動が抑えられたと考えている。

5. おわりに

本研究は EDCL を基盤にして様々な条件で準最高優先度を与えることによってデッドラインミスやコンテキストスイッチ、スケジューラの起動が従来と比較してどのように変化するか評価をした。この研究は飽くまでシミュレーションのため、スケジューリングアルゴリズムの評価として必要不可欠なスケジューリング可能性解析を行っていないため一般的な評価されているとは言えないため、今後はその解析を行っていきたいと考えている。

謝辞

研究を支援していただいた皆様に感謝いたします。

参考文献

- [1] 武田瑛, 加藤真平, 山崎信行, “Rate Monotonic に基づくマルチプロセッサ用のリアルタイムスケジューリング”, 情報処理学会論文誌コンピューティングシステム, Vol.2, No.1, pp.64-74(2009)
- [2] C.L.Liu and J.W.Layland, “Scheduling Algorithms for Multiprogramming in a Hard-Real Time Environment”, Journal of ACM, Vol.20, No.1, pp.46-61(1973)
- [3] S. Kato and N. Yamasaki : “Global EDF-based scheduling with laxity-driven priority promotion”, Journal of systems Architecture, Vol.57, No5, pp.498-571(2011)
- [4] 林 竜太, 兪 明連, 横山孝典, “マルチプロセッサにおけるリアルタイムスケジューリングアルゴリズムの提案”, 電子情報通信学会技術研究報告, Vol.115, No.374, pp.27-32(2015)