デッドラインを考慮したメモリ削減スケジューリングLMCLFの改善

組込みデザイン研究室　新井　諒介　(指導教員　中田　明夫)

## １　まえがき

スマートフォン,家電製品,医療機器などの組込みシステムは大量生産されることが多いため,製造コストの削減は重要な課題であり,メモリ消費量を削減することが,組込みシステムの開発目標の一つとして挙げられている[1].そこで我々は,マルチタスクシステムのヒープメモリ消費量を動的解析によって削減するスケジューリングアルゴリズムLeast Memory Consumption First (LMCF)スケジューリングを提案した[2]．さらに先行研究[3]では,メモリ消費量だけでなく,デッドライン制約も共に考慮したLeast Memory, remaining Computation-time, and Laxity First (LMCLF) スケジューリングが提案されている.LMCLFスケジューリングでは，設計者が任意に定めるパラメータαを導入し，単位メモリ消費量のα倍を単位時間の処理遅延と同等とみなすことでメモリ消費量とデッドライン充足の両者を勘案したスケジューリングを実現していた．しかし,メモリ消費量とデッドラインに大きなばらつきがある場合は，事前に適切なαの値を設定するのは困難である.もしαの値が最適でない場合,最適である場合よりもメモリ削減量が減ってしまう可能性がある.本研究では,そのαのより適切な値をスケジューラ動作中に自動推定する手法を提案する.提案手法により,αを事前に定める必要が無くかつ,αの値が最適でない場合の従来手法よりもメモリ消費量を削減することができる.

## ２　ヒープメモリと実時間制約を共に考慮したスケジューリング手法LMCLF

LMCLFスケジューリング[3]では，次のステップの式(1)のを計算し,が小さい から順に優先度を付与する.

## ３　提案手法

　従来では,式(1)のαの値が任意で与えられていて事前に求める必要があった.本研究では、そのαの値を任意に設定するより適切なαの値の導出法を提案する．提案手法をLMCLFスケジューリングに組み込むことで従来よりもメモリ消費量を削減することが可能となる．手法としてはまず,タスクの２ステップ後までのαの上限下限の値を求めその値が条件を満たすならばその時の最悪メモリ消費量を求めその時のαの上限下限を暫定のαの値とする.これを繰り返し行い最悪メモリ消費量が最小となるαを選択する.詳細は図１に示す.



図１:提案手法のアルゴリズム

## ４　実験

実験目的はαの値が最適でないとき提案手法が従来手法よりもメモリ削減されているかである.実験方法に関しては生成したタスクセットに対してスケジューリングを行い従来手法と提案手法の最悪メモリ消費量比較する.詳細については文献[3]と同様.

## 5あとがき

本研究では,[3]で提案されたLMCLFスケジューリングにおけるαの値のより適切な値の導出法を提案した.提案手法により,αを事前に定める必要が無くかつ,αの値が最適でない場合の従来手法よりもメモリ消費量を削減することができると考えられる.評価実験がまだ行えていないため結果はまだわからない.

## 参考文献

[1]R. Zurawski, “Embedded Systems Handbook, Second Edition: Embedded Systems Design and Verification”, CRC Press, 2009.

[2] Y. Machigashira and A. Nakata, “An improved LLF scheduling for reducing maximum memory consumption by considering laxity time”, In Proc. of 12th Int. Symp. on Theoretical Aspects of Software Engineering, pp.144–149, IEEE Computer Society Press, 2018.

[3] 町頭優輝, 中田明夫, 「ヒープメモリ確保・解放量と実時間制約を共に考慮しマルチプロセッサシステムのメモリ消費量を削減するリアルタイムスケジューリング」, 電子情報通信学会ソフトウェアサイエンス研究会報告(SS2019), 信学技報(SS2019–45), pp.25–30, 2020.

[4] T. P. Baker, “Comparison of Empirical Success Rates of Global vs. Partitioned Fix-Priority and EDF Scheduling for Hard Real Time”, Technical Report TR-050601, Department of Computer Science, Florida State University, pp.1–14, 2005.

[5] J. Lee, “Time-Reversibility for Real-Time Scheduling on Multiprocessor Systems”, IEEE Transactions on Parallel and Distributed Systems, Vol. 28, No. 1,