

B-031

**AnT** オペレーティングシステムにおける高精度な周期実行制御法の設計  
 Design of Sophisticated Periodic Execution for **AnT** Operating System

古川 友樹<sup>†</sup> 谷口 秀夫<sup>†</sup> 田端 利宏<sup>†</sup>  
 Yuuki Furukawa Hideo Taniguchi Toshihiro Tabata

1. はじめに

組込みシステムでは、リアルタイム性の要求が高い。このため、高精度なリアルタイム制御が可能なシステムが求められている。既存の代表的なオペレーティングシステムとして ART-Linux[1]がある。

本稿では、高精度な周期実行制御法について述べる。また、本機構を **AnT** オペレーティングシステム[2]に実装し、その評価結果を報告する。

2. 基本的な処理の流れと要求

2.1 基本的な処理の流れ

周期実行制御の基本的な処理の流れを図1に示し、以下に説明する。以降では、周期実行制御を行うプロセスを実時間プロセス、それ以外のプロセスを非実時間プロセスと名付ける。

周期実行では、実時間プロセスは非実時間プロセスとは別に管理される。登録された実時間プロセスは、実時間プロセスの READY キューへ繋がれ、優先度に基づいて選択され、実行される。待機を要求した実時間プロセスは、WAIT キューへ繋がれ、待機状態になる。この時、起動待ち時間が計算され、設定される。起動待ち時間はタイマ割り込み毎に減算され、起動待ち時間が0以下となった実時間プロセスは、READY キューへ繋がれる。削除を要求した実時間プロセスは、非実時間プロセスとして管理される。

2.2 要求

高精度な周期実行制御には以下の要求がある。

(要求1) タイマ割り込み処理における起動待ち時間計算の処理オーバーヘッドを抑制する。

起動する実時間プロセスに関する処理時間が長いと、他の実時間プロセスの実行が長く待たされるため、これを抑制する。

(要求2) 待機処理のオーバーヘッドを抑制する。

待機処理のオーバーヘッドが大きいと、次に実行待ちの実時間プロセスが長く待たされるため、これを抑制する。

(要求3) タイマ割り込みから実行までの処理時間の変動を抑制する。

タイマ割り込みから実行までの処理時間の変動は、周期実行の精度を低下させるため、これを抑制する。

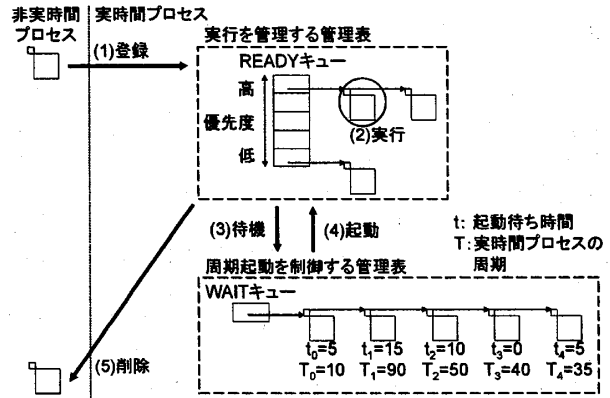


図1 周期実行制御の処理の流れ

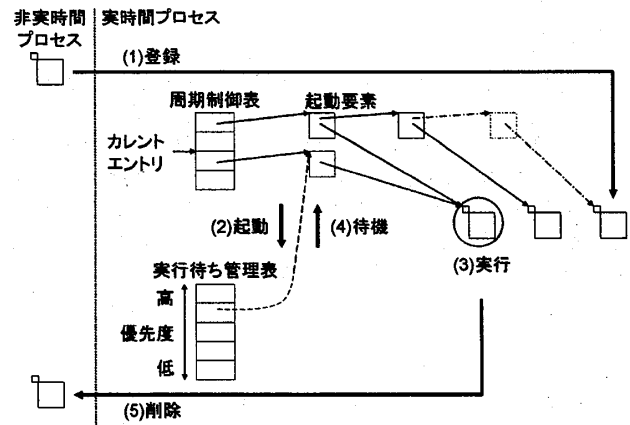


図2 高精度な周期実行制御法

3. 高精度な周期実行制御法

要求を満足する周期実行制御法の様子を図2に示し、以下に述べる。

(1) 登録

非実時間プロセスは、実時間プロセスへの登録を要求すると、実時間プロセスとして周期制御表により管理される。周期制御表は、要求される周期の最小公倍数分のエントリを持つ配列であり、周期に対応するエントリに実時間プロセスの情報を持つ要素(以下、起動要素)を繋ぐ。起動要素は、周期制御表のエントリの中で管理する起動要素の数が少ないエントリに繋がれる。この時、プロセス管理表に周期と優先度を設定する。

(2) 起動

タイマ割り込み発生時に、その時刻に対応する周期制御表のエントリ(以下、カレントエントリ)に起動要素が存在する場合、対応する実時間プロセスを起動する。具体的には、対応する実時間プロセスの優先度に合わせて、起動要素を実行待ち管理表に繋ぐ。

<sup>†</sup> 岡山大学大学院自然科学研究科 Graduate School of Natural Science and Technology, Okayama University

(3) 実行

実行待ち管理表に起動要素が存在する場合、対応する実時間プロセスを優先度順に実行する。この時、起動要素は、実行待ち管理表から外されない。

(4) 待機

実時間プロセスが待機を要求すると、対応する起動要素を実行待ち管理表から外す。これにより、実時間プロセスは、次の実行開始時刻まで待機する。

(5) 削除

実時間プロセスが削除を要求すると、対応する起動要素を周期制御表と実行待ち管理表から外す。また、登録の処理でプロセス管理表に設定した周期と優先度の値を初期化する。

以上のことから、周期制御表は周期毎に起動する実時間プロセスを管理するため、起動待ち時間計算を行う必要がない。このため、(要求 1)を満たすと推察できる。また、待機処理では、実行待ち管理表から起動要素を外すだけである。このため、(要求 2)を満たす。さらに、キューに繋がっている起動要素を探索する処理が少ないため、処理時間の変動が抑制できる((要求 3)を満足)と考える。

4. 評価

4.1 環境

提案した周期実行制御法を *AnT* オペレーティングシステムに実装し、周期実行の間隔について評価した。周期実行の間隔とは、各実行開始時刻と1周期前の実行開始時刻の差である。また、周期実行の間隔の誤差は、周期実行の間隔とプログラムが要求した周期の差である。測定は、表1の環境で行い、時刻の記録には RDTSC 命令を使用した。また、同じ内容について20回測定した。

4.2 考察

測定結果を図3と図4に示す。

(1) 図3より、周期と周期実行の間隔の誤差が比例している。この要因を明らかにするため、タイマ割り込みの周期の精度を測定した。測定の結果、周期 1ms に対し、実際のタイマ割り込みの間隔は 1.005ms であった。つまり、誤差が約 0.5%あり、この誤差の影響が要因といえる。なお、この誤差の原因として、システムが要求するタイマ割り込みの周期に対し、タイマの動作クロックの精度が一致しないため、正確な周期でタイマ割り込みが発生しないことが考えられる。

(2) 図3より、周期実行の間隔の最大値、最小値、および平均値の違いは非常に小さく、その変動は少ないといえる。図4においても周期の値に関係なく分散は  $0.0002 \mu s^2$  と小さく、このことを裏付けている。したがって、(要求 3)を満足していると推察する。

表1 測定環境

CPU	Intel Celeron D 2.8GHz
メモリ	256MB
タイマ割り込みの周期	1ms
外部計算機との接続	なし
登録可能な最小周期	1ms
登録可能な最大周期	100ms

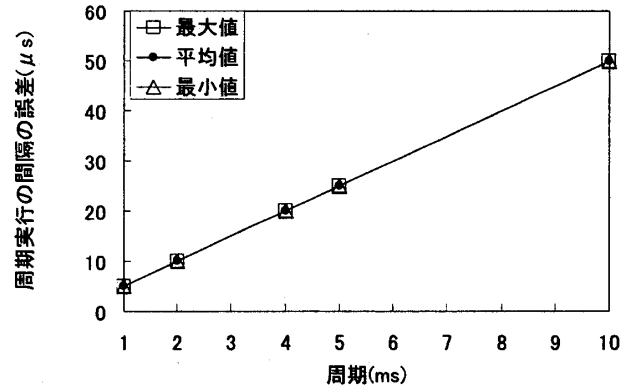


図3 周期実行の間隔の誤差

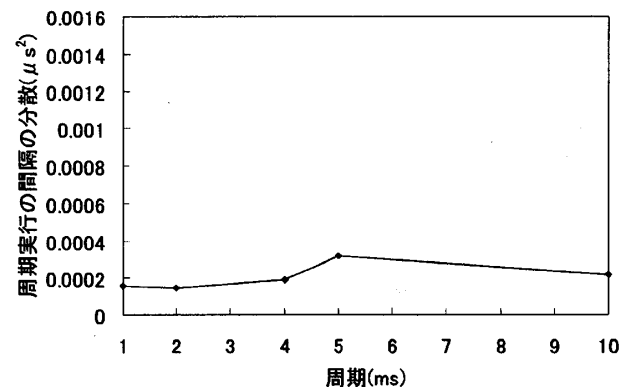


図4 周期実行の間隔の分散

5. おわりに

高精度な周期実行制御法について、基本的な処理の流れを述べ、制御法への要求を明らかにした。要求を満足する制御法を提案し、*AnT* オペレーティングシステムに実装して評価した。評価により、タイマ割り込みから実行までの処理時間の変動が抑制できることを示した。

残された課題として、タイマ割り込み処理における起動待ち時間計算の処理オーバーヘッドの抑制、および待機処理のオーバーヘッド抑制の定量的な評価がある。

謝辞 本研究の一部は、科学研究費補助金基盤研究(B)「適応性と頑健性を有する基盤ソフトウェアのカーネル開発」(課題番号: 18300010)による。

参考文献

[1]石綿 陽一, “SMPカーネルに基づく ART-Linux の安定化と実時間処理性能の測定”, 第3回計測自動制御学会システムインテグレーション部門講演会 論文集, Vol.si2002, pp.424-425(2002).  
 [2]谷口 秀夫, 乃村 能成, 田端 利宏, 安達 俊光, 野村 裕佑, 梅本 昌典, 仁科 匡人, “適応性と堅牢性をあわせ持つ *AnT* オペレーティングシステム”, 情報処理学会研究報告, 2006-OS-103, Vol.2006, No.86, pp.71-78(2006).