

周期実行制御における超過処理防止法の実現方式 Implementation of control method to deal with deadline miss for periodic scheduling

古川 友樹[†] 山内 利宏[†] 谷口 秀夫[†]
Yuuki Furukawa Toshihiro Yamauchi Hideo Taniguchi

1. はじめに

ロボットのモータ制御やセンサ制御における処理は周期的に実行される。周期実行制御を実現する方法として、様々なリアルタイムスケジューリングアルゴリズムが提案されている[1][2]。しかし、何らかの原因により、終了予定時刻までに処理が終了しない場合への対処は全くなされていない。例えば、処理の終了予定時刻超過時の対処として、超過を無視した処理の継続、および強制終了がある。あるいは、終了予定時刻超過が全く発生しないように厳格な実行管理を行う方法もある。しかし、処理を継続した場合、当該周期処理の次周期実行は正しい周期で実行されない問題がある。また、強制終了した場合、強制終了される周期処理が関連する処理へ悪影響を与えてしまう。

ロボット[3]などの組み込みシステムでは、より細かな制御を行うため、周期実行される処理の周期が短くなっている。また、実行する処理が高度化[4]しており、それに伴い機能拡充の要請が多い。したがって、超過処理を防止する対策が重要である。

我々は、処理終了予定時刻が近付くと、その旨を処理に通知し、終了予定時刻までに行える適切な処理を実行できる超過処理防止法[5]を提案した。ここでは、その実現方式を述べる。

2. 超過処理防止法

超過処理防止法では、処理の実行状況を把握し、終了予定時刻が近付くことを検知し、処理終了予定時刻を超過するか否かを判定し、超過しそうな場合は緊急処理を呼び出す。超過処理防止法の特徴は、以下の3つである。

- (特徴 1) 複数のモジュールに処理を分割し、処理の実行状況を把握する。
- (特徴 2) 正常終了するときのモジュールの実行時間を保存する。
- (特徴 3) 終了予定時刻を超過する場合、残り時間に対し、適切な処理を呼び出す。

基本的な処理方式を図1に示し、以下に説明する。処理を n 個のモジュールに分割し、処理の実行状況を管理する。モジュール間移行毎に、システムコールを用い、処理は OS に通知を行う。予め処理を試行し、処理から OS への通知があった時刻を記録する。終了予定時刻 T_i までに処理が終了した場合、記録した時刻から、各モジュール s_i の実行時間 $T(s_i)$ を算出し、実行ログを作成する。また、残り時間 U_i をユーザに通知する。処理が終了予定時刻 T_i を超過した場合、時刻 T_i における処理の実行状況と残り実行時間 V_i 、および実際の処理終了時刻 t_n をユーザに通知

[†] 岡山大学大学院自然科学研究科 Graduate School of Natural Science and Technology, Okayama University

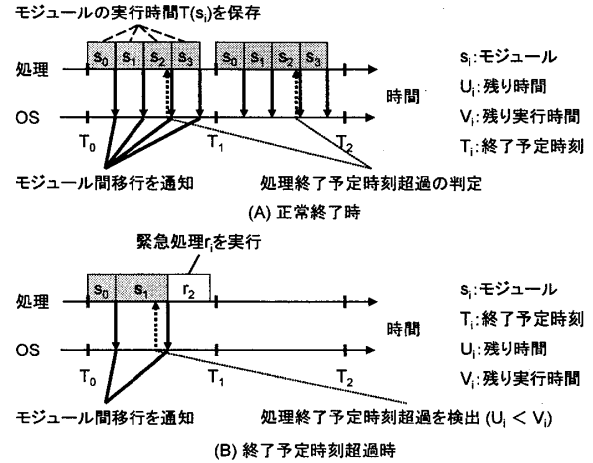


図1 基本的な処理方式

する。処理の終了予定時刻 T_i が近付いたとき、検出処理を実行する。検出処理において、処理終了予定時刻超過と判定された場合、緊急処理 r_i を呼び出す。

検出処理において、そのときの時刻 t_i と終了予定時刻 T_i から残り時間 U_i を算出し、実行ログのモジュールの実行時間 $T(s_i)$ から残り実行時間 V_i を求める。残り時間 U_i は判定時の時刻 t_i から次の起動時刻 T_i までの時間であり、残り実行時間 V_i は終了していないモジュールの実行時間 $T(s_i)$ 以上である場合、終了予定時刻までに処理を終了できると判定し、検出処理を終了する。一方、残り時間 U_i が残り実行時間 V_i より短い場合、処理終了予定時刻超過と判定し、残り時間 U_i に適切な緊急処理 r_i を実行する。

3. 実現方式

3.1 方針

超過処理防止法の実現における方針を以下に示す。

(方針 1) プロセスと OS の機能分担

プロセスに依存する処理の部分は、プロセスに処理させることで、プロセスと OS の間のやり取りを最小にし、オーバーヘッドを削減する。

(方針 2) 手続き部と管理データ部の分離

手続き部とデータの管理部を分離することで、周期実行されるプロセスが複数実行される機能への拡張を容易化する。

3.2 基本構造

基本構造を図2に示す。プロセスは、周期処理、動作記録処理、および緊急処理を有する。周期処理は、周期実行制御部により、周期実行制御される処理である。また、モジュール間移行時に移行検出部に通知を行う。動作記録処理は、モジュール管理部からモジュールの実行時間

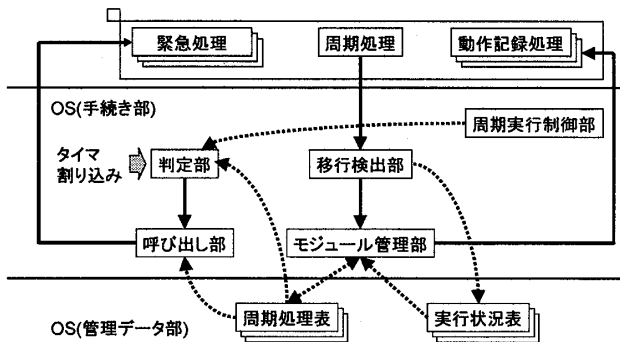


図2 基本構造

周期処理識別子(pid)
モジュール数:n
実行中のモジュール識別子
緊急処理のアドレス
モジュール1の実行時間
モジュール2の実行時間
...
モジュールnの実行時間

周期処理識別子(pid)
通知0の時刻
通知1の時刻
...
通知nの時刻

図3 管理表

を受け取り、保存する。緊急処理は、終了予定時刻を超過すると判定された際に呼び出される。判定部は、処理終了予定時刻が近付いたとき、処理終了予定時刻超過を判定する。呼び出し部は、処理終了予定時刻超過と判定された処理に対し、緊急処理を呼び出す。移行検出部は、周期処理のモジュール間移行を検知し、周期処理の実行状況を管理する。周期処理表と実行状況表の詳細を図3に示す。

3.3 処理の流れ

提案システムの4つの管理部の処理の流れについて、以下に説明する。

周期処理は、周期実行制御部により、周期実行される。このとき、周期処理は、モジュール間移行毎に、システムコールを発行し、移行検出部に通知する。移行検出部は、モジュール間移行を検出すると、モジュール管理部に制御を移行する。

モジュール管理部は移行検出部から制御を移行されると、周期処理表のデータを更新する。周期処理が周期実行を開始するとき、モジュール管理部は、周期処理表に周期処理のモジュール数、緊急処理部のアドレスを登録する。タイマ割り込み発生時、処理終了予定時刻に近い処理が存在すれば、判定部に制御が移行する。判定部は、周期実行制御部から終了予定時刻 T_i を取得し、判定時刻 t_i を差し引いた残り時間 U_i を計算する。また、周期処理表から終了していないモジュールの実行時間を取得し、残り実行時間 V_i を計算する。残り時間 U_i が残り実行時間 V_i 以上であれば、終了予定時刻までに処理が終了できると判定し、処理を終了する。そうでなければ、処理終了予定時刻超過と判定し、呼び出し部に制御を移行する。

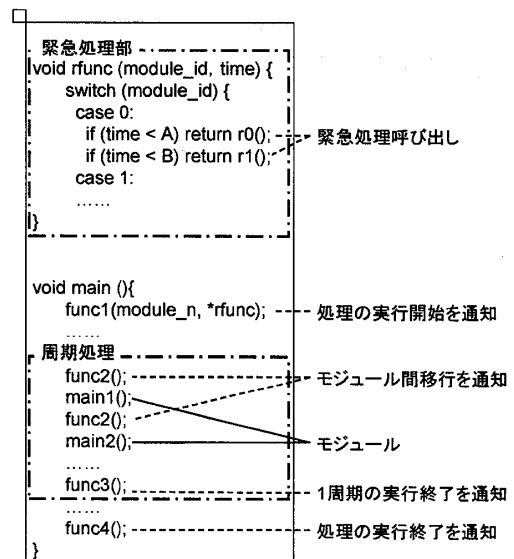


図4 処理の記述

呼び出し部は、周期処理表から緊急処理部のアドレスを取得し、緊急処理を呼び出す。緊急処理は、実行中のモジュール識別子と残り時間から適切な処理を選択し、実行する。

提案システムの処理の記述を図4に示す。周期処理開始前に、処理は $func1()$ を呼び出し、超過制御管理表に周期処理のモジュール数と緊急処理部のアドレスを登録する。1周期の処理終了後、 $func3()$ を呼び出し、超過制御管理表にモジュールの実行時間が記録されていない場合、実行状況管理表に記録された時刻から各モジュールの実行時間を計算し、超過制御管理表に保存する。また、残り時間をユーザに通知する。周期実行制御が終了すると、 $func4()$ を呼び出し、管理表から削除する。

検出処理において、残り時間より残り実行時間の方が長いとき、緊急処理部 $rfunc()$ を呼び出す。 $rfunc()$ が呼び出されると、実行中のモジュール($module_id$)と残り時間($time$)に応じた緊急処理を選択する。

4. おわりに

処理終了予定時刻が近付くと、その旨を処理に通知し、終了予定時刻までに行える適切な処理を実行できるシステムの設計について述べた。残された課題は、システムの実現と評価である。

参考文献

- [1] Liu C., Layland J., "Scheduling algorithms for multiprogramming in a hard real-time environment," Journal of the ACM, Vol.20, pp.46-61, (1973).
- [2] Giorgio C. Buttazzo, "Rate monotonic vs. EDF: judgment day," Real-Time Systems, The International Journal of Time-Critical Computing, Vol.29, Issue 1, pp.5-26, (2005).
- [3] Yokoi K., Kanehiro F., Kaneko K., Kajita S., Fujiwara K., Hirukawa H., "Experimental Study of Humanoid Robot HRP-1S," Intl. J. Robotics Research, Vol.23, No.4-5, pp.351-362, (2004).
- [4] Buttazzo G., "Research trends in real-time computing for embedded systems," ACM SIGBED Review, Vol.3, Issue 3 pp.1-10, (2006).
- [5] 古川友樹, 山内利宏, 谷口秀夫, "周期実行制御における処理終了予定時刻超過の検出と対処" 2010年並列/分散/協調処理に関する『金沢』サマー・ワークショップ, (2010).