

物理時間駆動と論理時間駆動に対応したリアルタイム OS

小林 万里[†] 横山 孝典[†] 兪 明連[‡]

東京都市大学[‡]

1. はじめに

複数の組み込みコンピュータをネットワーク接続した分散型組み込み制御システムは、実時間に基づいた処理が要求される。ジッタの少ないリアルタイム分散処理を実現する手法として時間駆動アーキテクチャ (Time-Triggered Architecture) があるが、その実現には通信時間が変動せず時刻同期機能が可能な時間駆動ネットワークを必要とする[1]。

我々は、ジッタが制御性能に影響する入出力タスクのみ同期した時間駆動(物理時間駆動と呼ぶ)で動作させ、ジッタが直接制御性能には影響しない算出タスクは仮想的な論理時間に基づいて非同期に実行する(論理時間駆動と呼ぶ)ことで、通信時間の変動を許容できる分散処理環境を提案した[2]。そして OSEK OS をベースとしたリアルタイム OS (RTOS) を使用し、物理時間駆動、論理時間駆動ともアラームを用いてタスク起動を行なった。しかしながら、時間駆動を実現するには AUTOSAR OS[3]のスケジュールテーブルのほうが適している。

そこで、本論文では、AUTOSAR OS を拡張し、システム時刻カウンタに関連付けたスケジュールテーブルを用いて物理時間駆動処理を、論理時刻カウンタに関連づけたアラームを用いて論理時間駆動処理を実現するリアルタイム OS を提案する。

2. 分散処理環境

2.1 論理時間を導入した時間駆動分散処理

論理時間を導入した時間駆動分散処理の動作を図 1 を用いて説明する。この例では、全てのタスクは周期 10 で動作し、2 つの算出タスクは論理時間駆動で動作している。論理時間とは、通信時間の変動を許容するために導入した仮想的な時間のことである。論理時間におけるタスクの起動時刻を論理起動時刻と呼ぶ。各タスクのジョブがメッセージを送信するとき、そのジョブの論理起動時刻をタイムスタンプとして付

加する。受信ノードでは、タイムスタンプに遅延時間を加えて、メッセージ受信により起動されるタスクの論理起動時刻を決定する。そして、入出力タスクはジッタの少ない高優先度の固定優先度タスク、算出タスクは応答時間を短縮するため EDF[4]でスケジューリングする。通信時間の変動によって算出タスクの起動時刻にずれが生じているが、論理的には時間駆動アーキテクチャと同等の動作と見なせる。

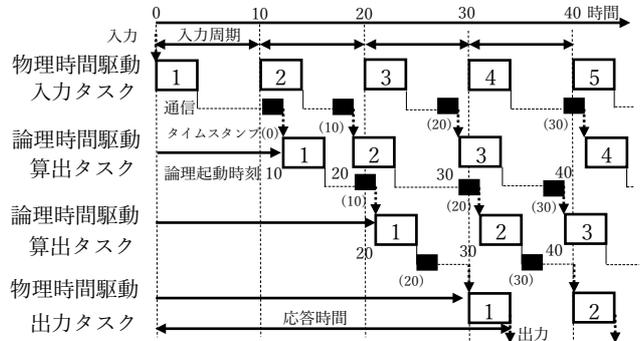


図 1: 分散処理環境の動作

2.2 リアルタイム OS

本 RTOS は AUTOSAR OS 仕様に基づく TOPPERS/ATK2[5]を拡張したもので、2.1 で述べた分散処理環境向けに、固定優先度スケジューリングと EDF スケジューリングの両者をサポートする。固定優先度で扱うタスクを固定優先度タスク、EDF で扱うタスクを EDF タスクと呼ぶ。

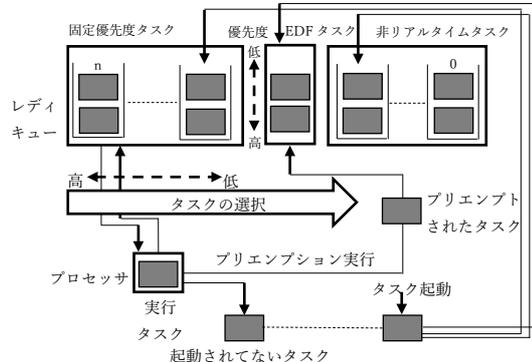


図 2: リアルタイム OS のスケジューリング機構

スケジューリングの動作について図 2 を用いて説明する。タスクが起動されると実行可能状態となりレディキューに格納される。レディキューは優先度ごとに存在し、固定優先度タスクは EDF タスクより高い優先度のレディキューに格納

A Real-Time OS Supporting Time-Triggered Processing Based on Physical Time and Logical Time

[†]Banri Kobayashi, Takanori Yokoyama and Myungryun Yoo

[‡]Tokyo City University

される。ただし、EDF タスクは一つのレディキューのみ使用し、デッドラインが近い順に格納する。また、それ以外の非リアルタイムタスクは EDF タスクより優先度が低いレディキューに格納される。タスクは優先度順に実行され、同じ優先度のタスクはレディキューの先頭から実行される。

3. タスク管理方式

3.1 タスクの起動

RTOS のシステム時刻はシステム時刻カウンタにより表現され、ハードウェアタイマによりティック周期で起動される割り込みで更新される。そして、図 3 に示すように、スケジュールテーブルをシステム時刻カウンタに関連づけ、そのスケジュールテーブルで入出力タスク（固定優先度タスク）を起動することで物理時間駆動を実現する。

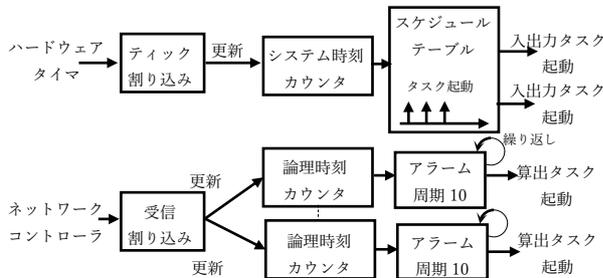


図 3: タスクの起動機構

一方論理時刻はメッセージ受信による割り込みで更新される論理時刻カウンタにより表現する。算出タスク毎に異なる論理時間を用いるため、論理時刻カウンタも論理時間毎に儲ける。そして、図 3 に示すようにアラームを論理時刻カウンタに関連づけ、そのアラームで算出タスク（EDF タスク）を起動することで論理時間駆動処理を実現する。

3.2 スケジュールテーブル

スケジュールテーブルは、時間駆動処理を実現することを目的に AUTOSAR OS に導入されたタスク起動方式である。スケジュールテーブルの動作について図 4 を用いて説明する。スケジュールテーブル周期はスケジュールテーブルが起動してから終了するまでの時間である。スケジュールテーブルはカウンタに接続して駆動し、スケジュールテーブル時刻に基づいてタスクの起動を行う。



図 4: スケジュールテーブルの動作

3.3 論理デッドラインに基づく EDF

論理時間駆動のタスクは論理デッドラインを使用して EDF スケジューリングする。タスク起動時に論理起動時刻にタスク毎に初期設定された相対デッドラインを加算して論理デッドラインを算出する。論理起動時刻はタスクを起動したアラームに接続されたカウンタの値で表現される。図 2 の EDF タスク用のレディキュー内は実行中のタスクを除いて最も論理デッドラインの近いタスクが先頭となる。

4. 実装と評価

実装にはプロセッサ V850E2/FG4-L を搭載している評価ボードを用いる。プロセッサの動作クロックは 80MHz で、内蔵メモリの容量は ROM が 1MB で RAM が 64B である。

タスクを起動してから動作が開始するまでの時間であるタスク起動時間の測定結果を表 1 に示す。TOPPERS/ATK2 と比較して処理の追加による起動時間の増加は数%程度であり、実用上問題ない範囲であると考えられる。

表 1: 起動時間の測定結果

OS	起動方法	対象タスク	起動時間 [μ sec]
TOPPERS/ATK2	スケジュールテーブル	固定優先度	2.43
	アラーム	固定優先度	2.25
本リアルタイム OS	スケジュールテーブル	固定優先度	2.46
	アラーム	固定優先度	2.28
		EDF	2.44

5. おわりに

本論文では、論理時間を導入した時間駆動分散処理環境向けのリアルタイム OS を提案した。今後は論理時間駆動もスケジュールテーブルに対応する予定である。

謝辞

本研究で使用した TOPPERS/ATK2 の開発者に感謝する。本研究は JSPS 科研費 JP18K11225, JP21K1815 の助成を受けたものである。

参考文献

- [1]. Kopez, H., Should Responsive Systems be Event-Triggered or Time-Triggered?, IEICE Trans. Inf. & Syst., Vol.E76-D, No.11, pp.1325-1332(1993)
- [2]. Ichimura, A. et al., A Time-Triggered Distributed Computing Environment for Cyber-Physical Systems Based on Physical Time and Logical Time, Proc. of IEEE TENCON2018, pp.1516-1521 (2018)
- [3]. AUTOSAR, Specification of Operating System Release4.0 Revision3(2011)
- [4]. Liu C. and Layland J.W., Scheduling Algorithms for Multiprogramming in a Hard-RealTime Environment, Journal of the ACM, Vol.20, No.1, pp.46-61(1973)
- [5]. TOPPERS/ATK2, <https://www.toppers.jp/atk2.html>