

マルチコアプロセッサ向け実行制御機能を有するリアルタイム OS A Real-Time Operating System with Execution Control Functions for Multi-Core Processors

杉本 樹生[†] 兪 明連[†] 横山 孝典[†]
Tatsuki Sugimoto Myungryun Yoo Takanori Yokoyama

1. はじめに

近年、組み込み制御システムに搭載されるソフトウェアが大規模化し、マルチコアプロセッサによる性能向上が図られている。組み込み制御分野では非対称型のマルチコアプロセッサ（AMP）が主流である。また最近では、消費電力を削減するため、動作コア数の変更やコア毎に動作周波数を変更するなど、より高度な管理が求められている。そのためにはマルチコアプロセッサにおいてコア毎の実行制御が可能なりリアルタイム OS（RTOS）が必要になる。

非対称型マルチプロセッサ向けに μ ITRON 仕様^[1]を拡張した TOPPERS/FDMP カーネル^[2]が提案されている。しかし、これはタスクやセマフォ等を対象としたシステムコールのみを提供しており、コア毎に RTOS の起動や停止をすることは想定していない。

OSEK/VDK 仕様^[4]を拡張とした自動車制御向け RTOS として AUTOSAR OS^[4]があり、マルチコアプロセッサ向けに拡張した仕様もある。マルチコア向け AUTOSAR OS 仕様では、コアの起動を行う StartCore というシステムコールを規定しているが、コア毎に RTOS の起動や停止をすることは想定していない。

我々は、より柔軟な実行制御を可能にすることを目的に、実行中に特定のコア上の RTOS の起動や停止を可能とするシステムコールを持つマルチコア向け RTOS を提案する。また、複数のコア上のアプリケーションが同一のシステム時刻に基づいて高精度で同期して処理を実行することを可能とするため、各コア上で同期した RTOS のシステム時刻を提供する。

2. RTOS

2.1 概要

我々はこれまでに、OSEK OS 準拠の TOPPERS/ATK1^[5]をベースにマルチコア並列処理と分散処理の両者に対応した RTOS^[6]を開発しており、それを拡張して目的とする RTOS を実現する。本 RTOS の構造を図 1 に示す。これまでに OSEK OS のタスク管理機能とイベント制御機能について、マルチコア向けのコア間遠隔システムコールと分散環境向けのノード間遠隔システムコールを実現している。これに本論文で提案する RTOS の実行制御に関わる機能を追加する。

2.2 実行制御機能

実行制御機能として任意のコア上の RTOS を起動、停止するための StartCoreOS, ShutdownCoreOS というシステムコールを追加する。StartCoreOS, ShutdownCoreOS は引数により起動・停止対象の CPU コアの CPU ID を指定する。

[†] 東京都市大学

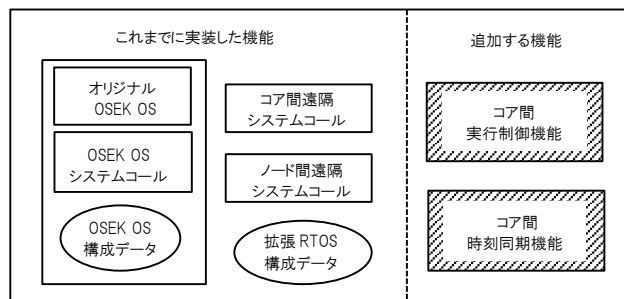


図 1 RTOS の構成

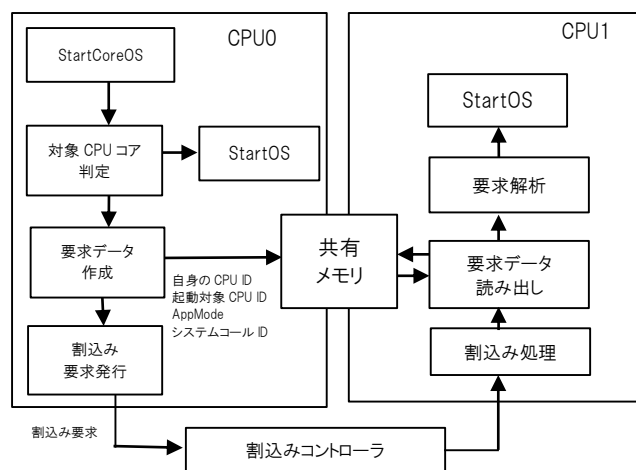


図 2 StartCoreOS の動作

また、StartCoreOS は OSEK OS の StartOS と同様に引数でアプリケーションモードを指定することができる。

図 2 に StartCoreOS の動作を示す。CPU0 で StartCoreOS を呼び出すと、対象コアが自身であるかを CPU ID により判定する。自身が対象の場合、OSEK オリジナルの StartOS を実行し、RTOS を起動する。他コアが対象の場合は、要求データとして要求元（自身）の CPU ID、起動対象の CPU ID、及びシステムコールの種類を表すシステムコール ID、アプリケーションモードを共有メモリに書き込み、対象コア（この場合は CPU1）に割り込み要求を出す。対象コア（CPU1）の割り込み処理は、共有メモリに書き込まれている要求データを読み出して解析し、対象が自コアでシステムコール ID が StartCoreOS を表すことを確認し、オリジナルの StartOS を実行する。ShutdownCoreOS は、StartOS の代わりに ShutdownOS を呼び出すことを除き、上記と同様の流れの処理を実行する。

2.3 時刻同期の方法

各 CPU コア上のタスクを同一のシステム時刻に基づいてスケジューリングするため、各 CPU コア上の RTOS のシステム時刻を一致させる。RTOS のシステム時刻はティック

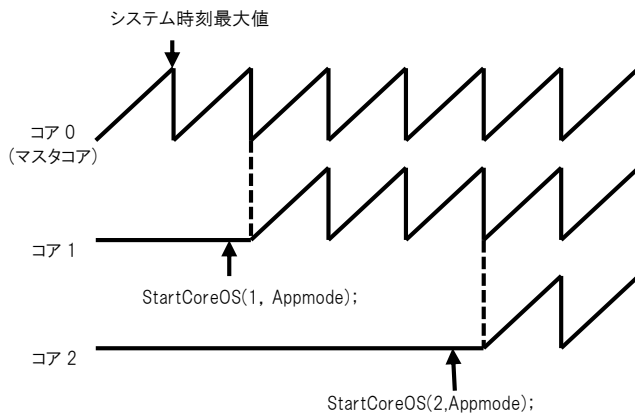


図 3 時刻同期機構

間隔で発生するハードウェアタイマの割込み処理によって更新され、最大値になると 0 にリセットされる。

図 3 を用いてシステム時刻の同期方法を説明する。最初に立ち上がった CPU コアをマスタコアとし、それ以外のコアはスレーブコアとなる。マスタコア上の RTOS はシステム時刻を更新するたびに共有メモリにその値を書き込む。StartCore-OS によりスレーブコアが起動されると、StartOS 内の初期化処理後に時刻同期を行う。具体的には共有メモリからシステム時刻を読み込み、マスタコアの時刻が 0 になるまで待機する。0 になったタイミングで待機を解除して RTOS を起動し、ハードウェアタイマをスタートさせ、システム時刻のカウントを開始する。システム時刻の値そのものをのみでなく、ハードウェアタイマのカウント値も一致させる。

3. 実装と評価

3.1 実装

本 RTOS の実装に使用した環境を表 1 に示す。デュアルコアプロセッサ SH7205 を搭載した評価ボード M3A-HS50 を使用する。SH7205 の各 CPU は両コアからアクセス可能な RAM を内蔵しており、CPU0 の内蔵 RAM を共有メモリとして使用する。

表 1 実装環境

CPU	SH2A-FPU(200MHz)×2
内蔵メモリ	CPU0: 高速内蔵 RAM (64K バイト) CPU1: 高速内蔵 RAM (32K バイト)
外部メモリ	フラッシュメモリ (64M ビット)
開発環境	High-performance Embedded Workshop V.4.09.01
コンパイラ	SuperH RISC engine C/C++コンパイラ パッケージ V.9.04 Release 03

3.2 実行制御機能の評価

本 RTOS のマルチコア管理機能が自動車制御などの組込み制御システムに適用可能かどうか、システムコールの CPU 実行時間を計測して評価する。StartCoreOS, ShutdownCoreOS の実行時間を表 2 に示す。他コアを対象にした

表 2 : 実行制御機能の評価結果

対象 システム コール	他コアを対象		自コア を対象	TOPPERS/ ATK1 オリジナル
	発行元 コア	発行先 コア		
StartCoreOS	9.84	78.48	74.20	72.46
ShutdownCoreOS	9.35	25.44	18.72	72.46

[μ sec]

場合と自身のコアを対象とした場合について測定し、他コアを対象とした場合については発行元コアの実行時間と、発行先コアの実行時間を分けて記載している。示した値は 100 回計測した平均値である。比較のため、オリジナルの TOPPERS/ATK1 の StartOS と ShutdownOS の実行時間も示した。

両システムコールとも、他コアを対象とした場合の発行元コア・発行先コア合計の実行時間は、自コアを対象とした場合より 20% ほど大きい。また、自コアを対象としたものもオリジナルの StartOS と ShutdownOS よりやや大きい。しかし、これらは RTOS の起動・停止時のみ実行するものであるため、上記程度の実行時間の増大は実用上問題ないと思われる。

3.3 時刻同期機能の評価

コア間時刻同期機能の評価のため、時刻同期の精度を測定する。各コアのシステム時刻更新割込み処理の起動に用いるハードウェアタイマのカウント値を比較し、観測時には常に一致していることを確認した。今回使用したハードウェアタイマは 1 カウント 0.24μs のため、0.24μs 以下の誤差で同期していると考えられる。本 RTOS のタスク起動にかかる CPU 実行時間の 2.4μs と比較して、時刻同期の誤差は十分に小さく、スケジューリングに与える影響はほとんどない。したがって実用上十分な精度で時刻同期を実現できたと考えている。

4. おわりに

各コア上の RTOS のシステム時刻を同期させながら起動・停止を行うシステムコールを持つ、非対称型マルチコアプロセッサ向けの RTOS を開発した。そして、それらのシステムコールの実行時間が実用上問題ない値であることを、及びシステム時刻を十分な精度で同期ができていることを確認した。

謝辞

本研究で使用した TOPPERS/ATK1 の開発者に感謝する。本研究は JSPS 科研費 JP15K00084 の助成を受けたものである。

参考文献

- [1] トロン協会 ITRON 仕様検討グループ, 坂村 健 (監修), 高田 広章 (編), “μ ITRON4.0 仕様 Ver.4.02.00”, 2008.
- [2] 本田晋也, 高田広章, “ITRON 仕様 OS の機能分散マルチプロセッサ拡張”, 電子情報通信学会論文誌, Vol. J91-D, No. 4, pp.934-944, 2008.
- [3] OSEK/VDX, “OSEK/VDX Operating System”, Version 2.2.3, 2005.
- [4] AUTOSAR, “Specification of Operating System”, Release 4.3.0 2016
- [5] TOPPERS/ATK1 カーネル, <http://www.toppers.jp/atk1.html>
- [6] 横山幸太郎, 齊藤政典, 兪明連, 横山孝典, “マルチコア並列処理・分散処理統合機能を有するリアルタイム OS”, 情報処理学会研究報告, Vol.2015-EMB-36, No.36, pp.1-6, 2015