

# マルチコアプロセッサを含む分散型組込みシステム向け リアルタイムOS

細山 博史<sup>†</sup> 横山 幸太郎<sup>†‡</sup> 兪 明連<sup>†</sup> 横山 孝典<sup>†</sup>  
<sup>†</sup>東京都市大学 <sup>‡</sup>現在, 株式会社日立製作所

## 1. はじめに

自動車制御等の分野において複数ノードをネットワーク接続した分散制御システムが用いられている。また近年、組込みシステム分野におけるマルチコアプロセッサへの注目が高まっており、特に非対称型マルチプロセッサの実用化が進んでいる。以下、本論文ではマルチコアプロセッサによる処理を並列処理、ネットワーク接続された複数ノード間による処理を分散処理と呼ぶことにする。

組込みシステムのソフトウェア開発にはリアルタイムOS(Real-Time Operating System, RTOS)を用いるのが一般的であり、並列処理及び分散処理の両者に対応可能なRTOSが求められている。

非対称型マルチコアプロセッサ向けのRTOSとして、μITRON4.0仕様を拡張したTOPPERS/FDMPカーネルがあり、異なるCPUコア間で位置透過性のあるタスク管理やタスク間同期を可能としている[1]。

分散制御システム向けのRTOSとして、FlexRayネットワークを対象に我々が開発した分散RTOSがあり、異なるノード間で位置透過性のあるタスク管理及びイベント制御を可能としている[2]。

さらに我々は、並列処理と分散処理が混在する環境において、同一APIを用いて統合的に管理できるRTOSを開発中である[3]。これを本論文では統合RTOSと呼ぶ。本RTOSではネットワークとしてCAN(Controller Area Network)を用いるが、既に発表したバージョンでは1つのCANコントローラを1つのCPUコアに対応させることで、全CPUコアにネットワーク通信機能を持たせた。本論文では、CANコントローラ搭載数がCPUコア数より少ない場合にも対応可能とするため、1つのCPUコアのみにネットワーク通信機能を持たせる構成とした統合RTOSについて述べる。

## 2. 統合RTOSの仕様

### 2.1 拡張対象システムコール

本研究ではOSEK OS仕様[4]を拡張し、マルチプロセッサにおける他CPUコア上や分散システムにおける他ノードのCPU上のタスクについて、自CPU上のタスクと同一のシステムコールを用いて管理可能とする。これにより、対象タスクがどのノードのどのCPUコア上に存在するかを意識せずにアプリケー

A Real-Time Operating System for Distributed Embedded Systems with Multicore Processors

<sup>†</sup>Hiroshi Hosoyama, Kotaro Yokoyama, Myungryun Yoo, Takanori Yokoyama, Tokyo City University

<sup>‡</sup>Presently with Hitachi, Ltd.

表1 タスク管理とイベント制御のシステムコール

分類	システムコール名(引数)	タスク指定
タスク管理	ActivateTask(Task)	○
	TerminateTask()	×
	ChainTask(Task)	○
	Schedule()	×
	GetTaskID(TaskIDRef)	×
	GetTaskState(Task,status*)	○
イベント制御	SetEvent(Task,Event)	○
	ClearEvent(Event)	×
	GetEvent(Task,Event)	○
	WaitEvent(EventMask*)	×

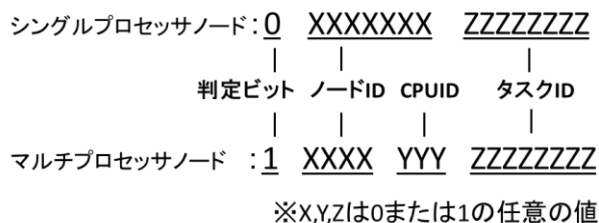


図1 グローバルタスクIDの割り当て

ションを記述できるようになる。

OSEK OS仕様が規定しているタスク管理、イベント制御、アラーム管理、リソース管理、割込み制御、OSの実行管理の6つのカテゴリのシステムコールのうち、タスクを対象としたタスク管理とイベント制御のカテゴリに属するシステムコールを表1に示す。タスク指定という欄は引数で対象タスクを指定するかどうかを示し、それに該当するActivateTask, ChainTask, GetTaskState, SetEvent, GetEventの5つのシステムコールを拡張し、他CPUコア上や他ノード上のタスクを指定可能とする。以下、他ノード上のタスクに対するシステムコールをノード間遠隔システムコールと呼ぶ。

### 2.2 グローバルタスクID

システム全体でタスクを一意に指定可能とするため、2ByteのグローバルタスクIDを定義し、図1に示すように上位1ByteでノードやCPUコアを表し、下位1ByteでOSEK OS本来のタスクIDを表す。上位1Byteのうち最上位ビットはシングルコアかマルチコアを表す判定ビットとし、マルチコアの場合は残りの上位4ビットでノードID、下位3ビットでCPUIDを表し、シングルコアの場合は全7ビットでノードIDを表す。

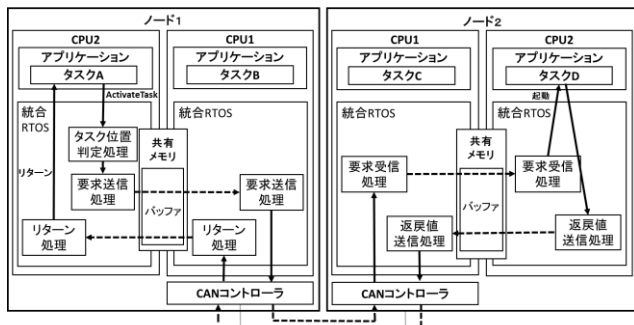


図2 ノード間遠隔システムコールの流れ

### 3. 統合 RTOS の機能と構成

#### 3.1 実装方針

本統合 RTOS はオープンソースの TOPPERS/ATK1 [5] を拡張して実装する. 実装に用いるハードウェアはルネサスのデュアルコアプロセッサ SH7205 を搭載した評価ボード M3A-HS50G50 である. SH7205 は共有メモリにより CPU コア間でデータを共有でき, また CAN コントローラを 2 つ搭載している. SH7205 の CPU コアのクロック周波数は 200MHz, CAN の通信速度は 500kbps とする.

#### 3.2 ノード間遠隔システムコール

本統合 RTOS では, ネットワーク通信機能を持たない CPU コアは, ネットワーク通信機能を持つ CPU コアに共有メモリ経由で要求を出すことで, 他ノードへの通信を実現する. 具体的には, 図 2 に示すデュアルコアプロセッサノードの CPU1 に CAN 通信機能を付与する. CAN 通信機能を有するかどうかは, システム構成情報を表すコンフィギュレーションデータを参照して判定する.

ノード間遠隔システムコール処理の流れについて図 2 を用いて説明する. ノード 1 の CAN 通信機能を持たない CPU2 上のタスク A がシステムコールを発行すると, 共有メモリを介して CAN 通信機能を有する CPU1 に要求送信データを転送した後, CPU1 が CAN 経由で要求送信を行う.

要求データを受信したノード 2 の CPU1 では, 受信処理により要求送信先の CPU コア判定を行う. 要求送信先が CPU1 の場合は, 図では省略しているが CPU1 でシステムコール処理を行う. 要求送信先が CPU2 の場合は, 要求送信データを共有メモリ経由で転送後, CPU2 でシステムコール処理を行う.

システムコール実行完了後, CPU2 は戻り値送信処理により戻り値を共有メモリ経由で CPU1 に転送し, CPU1 は戻り値を CAN 経由でノード 1 に送信する. ノード 1 の CPU1 は戻り値を受信後, リターン処理により戻り値を共有メモリ経由で CPU2 に転送し, CPU2 上の統合 RTOS がシステムコール発行元のタスク A に戻り値をリターンする.

表 2 システムコールの実行時間 ( $\mu$  sec)

システムコール	要求送信	要求受信	戻り値送信	リターン処理
ActivateTask	10.1	8.3	4.5	8.6
ChainTask	10.8	8.3	4.5	9.3
GetTaskState	10.7	8.2	4.5	9.1
SetEvent	10.1	8.3	4.5	8.6
GetEvent	10.5	8.2	4.6	10.4

### 4. 評価

本研究で実装した統合 RTOS のノード間遠隔システムコールの実行時間を測定した. 共有メモリ経由転送処理を含む場合のノード間遠隔システムコールの実行時間を表 2 に示す. 表に示した値は CPU1 及び CPU2 の合計処理時間である. これらに CAN の通信時間を加えたものがノード間遠隔システムコールの応答時間となる. 通信速度 500kbps の場合, 1 回の CAN 通信に 0.2msec 程度要するが, それと比較して CPU 実行時間は十分小さい値である. また, CAN 通信時間を含めたノード間遠隔システムコールの応答時間は 0.5msec 程度となるが, 自動車制御アプリケーションにおいて分散処理を行うタスクの制御周期は 10msec から 100msec 程度であるため, 実用上は問題ないと考えている.

### 5. おわりに

並列処理と分散処理が混在する環境において統一的なタスク管理を可能とする統合 RTOS を提案するとともに, 実装及び評価を行い, 実用上問題ない性能であることを確認した.

現在, 統合 RTOS に対応したコンフィギュレーションデータを生成可能なシステムジェネレータを開発中である.

### 謝辞

本研究で使用した TOPPERS/ATK1 の開発者に感謝する. 本研究費は JSPS 科研費 JP2450046 および JP15K00084 の助成を受けたものである.

### 参考文献

- [1] 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)
- [2] 知場貴洋, 齊藤政典, 伊丹悠一, 兪明連, 横山孝典: 位置透過性のあるシステムコールを有する組み込み制御システム向け分散リアルタイム OS, 情報処理学会論文誌, Vol. 53, No. 12, pp. 2702-2714 (2012)
- [3] 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)
- [4] OSEK/VDX: OSEK VDX Operation System, Version 2.2.3 (2005)
- [5] TOPPERS プロジェクト: TOPPERS/ATK1, <http://www.toppers.jp/atk1.html>