

# SMP方式「Try Kernel-S」の概要と実装

[ご購入はこちら](#)

豊山 祐一

第5部ではSMP (Symmetric Multi-processor, 対称マルチコア制御) 方式のリアルタイム OSを作ります。

SMP方式は全てのCPUコアが対等に動作し、特定の役割が定められていません。SMPに対応したプログラムはどのCPUコアでも実行が可能であり、実行時にどれかのCPUコアが選ばれて処理されます。そのため、PCやスマートフォンなどのプロセッサでは標準の方式です。一方で組み込みシステムでは、OSにLinuxなどを使用するような製品で使われることもあります。マイコンを使用した製品ではあまり一般的ではありません。

しかし、SMP方式のOSを作成することにより、その仕組みが理解できると同時に、組み込みシステムにおけるSMP方式のメリット、デメリットもはっきりしてくるのではないかと思います。

SMP方式をTry Kernelに対応させるにあたり、Try Kernel 2.0をベースとし、名前はTry Kernel-Sとします。

## 設計の方針とシステムの構成

SMP方式ではCPUコアの数に関係なく、1つのOSがシステムを管理します。OS上で実行されるアプリケーション・プログラムも1つです。

アプリケーション・プログラムはマルチコアをできる限り意識せずに、理想的にはシングルコアのOSのアプリケーション・プログラムと同一のプログラムが実行できることが望ましいです。そこでTry Kernel-SのAPIの仕様は、シングルコア版Try Kernelと基本的に同じとします。

タスクのスケジューリングは、OSの設計において、OSの特徴や性能、アプリケーション・プログラムの動作を決める重要な要素です。そこでTry Kernel-Sのタスクのスケジューリングについては第2章で詳しく説明します。

Try Kernel-Sのシステム全体のソフトウェア構成を図1に示します。OSもアプリケーション・プロ

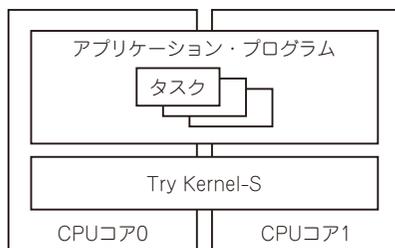


図1 Try Kernel-Sのソフトウェア構成概要

ラムもCPUコアを共有し、アプリケーション・プログラムのタスクは特定のCPUコアには依存しません。そのためシングル・コア版のTry Kernelとよく似た構成になります。ただし、OS内部の構成や動作は、複数のCPUコアの制御を行うためにシングル・コア版とは異なります。

## 設計ポイント① 内部構造

シングルコア版Try Kernelを基にTry Kernel-Sの内部設計を行っています。OSを構成する内部の各プログラムについてSMP化を検討します。

### ● API処理部

API処理部の実体はアプリケーション・プログラムのタスクから呼び出されるC言語の関数です。よってAPI処理部は呼び出し元のタスクと同じCPUコアで実行されます。

APIの機能は基本的にシングルコアのTry Kernelと同じですので、APIの処理自体にSMP化による変更は基本的にはありません。

### ● 時間管理部

時間管理部は、ハードウェアのタイマ割り込みを使って時刻を計測し、OSの時間管理を行います。時間管理部の実体は、SysTickタイマの割り込みハンドラです。

SysTickタイマはCPUコアごとにありますが、OS