第3章 機能検討から生成APIの設計，超音波距離センサを使った実験まで

CPUコア間メッセージの実装

豊山 祐一

AMP方式では，各CPUコアで実行されるプログラムの独立性が高いほど，CPUコア間の干渉が減少し，システム全体の実行効率が向上します．リアルタイムOSの観点でも，タスクの時間予測性と応答性能を向上させるためには，CPUコア間のプログラムの独立性が重要です．

しかし，各CPUコアのプログラムが互いに無関係と言うことはあまりありませんので，完全な独立性を実現することは困難です．ただし，CPUコア間の通信に非同期性の高い通信方法を採用することで，独立性を向上させることはできます．

そこで，Try Kernel-AのCPUコア間の通信機能として，非同期性が高いメッセージ通信機能を実装しました．

CPUコア間メッセージの設計方針

リアルタイムOSの国際標準規格IEEE 2050-2018では，タスク間のメッセージ通信の機能として，メッセージ・バッファが規定されています．Try Kernel 2.0でもメッセージ・バッファを実装しています．メッセージ・バッファについては特設記事で説明しています．本章はメッセージ・バッファについて理解していることを前提として説明します．

ただし，IEEE 2050-2018規格にはマルチコアに関する規定はありません．そこで，メッセージ・バッファの機能を基にして，CPUコア間でのメッセージ機能を設計します．これには主に次の2つの案が考えられます．

- 案1：既存のメッセージ・バッファ機能を拡張して，CPUコア間の通信を可能にする
- 案2：CPUコア間メッセージ機能を新規に実装する

案1は技術的には整合性が取りやすく，既存のAPIをそのまま活用できるため，プログラマがCPUコア間の特殊な通信を意識せずに済むというメリットがあります．しかし，この案を実現するためには，既存のメッセージ・バッファ機能をCPUコア間に拡張する際に比較的大きなコード変更が必要となります．

一方，案2では，必要な機能のみを新規に実装することで，設計と実装の両面でシンプルさが保たれます．また，AMP方式のシステムを構築する上では，CPUコア間の通信を意識する必要があることから，新しい機能を明確に区別することが望ましいことも考えられます．

以上の理由から，Try Kernel-Aでは案2を採用することとしました．

CPUコア間メッセージの機能

CPUコア間メッセージの基本的な機能を，メッセージ・バッファの機能を元に次のように定めます．

- CPUコア間メッセージはTry Kernel-Aのカーネル・オブジェクトとして実装します．セマフォやイベント・フラグなど他のカーネル・オブジェクトと同じように，APIで生成し制御を行います．
- CPUコア間メッセージは，異なったCPUコアで実行されているタスク間でメッセージの通信を行う機能を提供します．
- メッセージは任意の内容のバイト列のデータとします．メッセージの最大サイズはCPUコア間メッセージの生成時に指定します．
- タスクから送信されたメッセージはCPUコア間メッセージに格納されます．CPUコア間メッセージに空きがない場合は，タスクは送信できるまで待ち状態となります．
- タスクは，CPUコア間メッセージに格納されたメッセージを受信できます．CPUコア間メッセージにメッセージが格納されていない場合は，タスクは受信できるまで待ち状態となります．

CPUコア間メッセージの実装を簡単にするため，メッセージはCPUコア間の通信にのみ使えるとし，同一のCPUコアで動作しているタスク間の通信には使えないこととしました．

つまり，CPUコア間メッセージでは**図1**(a)に示す