第5部

第2章 SMP方式の要であるスケジューラ&ディスパッチャを実装する

タスク・スケジューリングの概要と実装

豊山 祐一

第2章では，Try Kernel-Sのタスクのスケジューリングについて説明します．タスクのスケジューリングは，マルチタスクのアプリケーション・プログラムの動作に直接影響を与えますのでOSの設計で重要なポイントとなります．

Try Kernel-Sのスケジューリングは，シングルコア版のTry Kernelの優先度スケジューリングのシンプルな拡張です．これに基づくスケジューラとディスパッチャの詳細を説明します．

設計ポイント④ スケジューリング規則

● スケジューリング規則

Try Kernel-Sのタスクのスケジューリングは，実行可能なタスクの中で優先順位の高い順にCPUコアの数だけタスクを実行するという方針です．

この方針に沿って具体的なタスクのスケジューリング規則を次のように定めます．

規則1：実行可能なタスクの中で，タスクの優先度が高いものほど優先順位が高いとする．また，同一の優先度のタスクでは先に実行可能な状態となったものほど優先順位が高いとする．

規則2：規則1の優先順位の高いタスクから順番に，CPUコア数の数だけのタスクを実行状態とする．

● タスクの優先度と優先順位は異なる

タスクの優先度と優先順位の違いに気を付けてください．

タスクの優先度は，タスクの生成時に指定したタスク固有の属性値の1つです．Try Kernelではタスクの優先度は生成時に決まった後は変更できません．また，同じ優先度のタスクが存在することもあります．

タスクの優先順位は，スケジューラが規則1に従って決めたタスクを実行する順番です．タスクの優先順位はその時点でのタスクの数や状態により変わります．

設計ポイント⑤ レディ・キュー

OSの中でタスクの優先順位を現したデータがレディ・キューです．レディ・キューは，タスクを優先度と時系列で並べた待ち行列であり，TCB（タスク制御ブロック）の双方向リストとして実装されます．

Try Kernel-Sのスケジューラは，CPUコア数に関係なくシステム全体としてタスクの優先順位を決めてますので，レディ・キューの構造はシングルコア版のTry Kernelと同一となります．

設計ポイント⑥ スケジューラ

● スケジューラの処理の流れ

スケジューラの処理の流れを**図1**に示します．

▶①実行するタスクの選択

スケジューラは，レディ・キューを調べて実行するタスクを選択します．

レディ・キューには優先順位の順番にタスクが並んでいます．シングルコアの場合はレディ・キューの一番先頭のタスクが実行するタスクでしたが，SMP方式ではCPUコアの数だけ実行するタスクが決まるまで，レディ・キューをたどっていきます．Picoの場合はデュアルコアですので，一番優先順位の高いタスクと2番目に優先順位の高いタスクが選択されます．

図2にレディ・キューから実行するタスクを選択する例を示します．

図2(a)では，最高優先度のタスクのレディ・キューから，先頭から2つのタスクが選ばれています．

図2(b)では最高優先度の実行可能なタスクが1つだけだったので，次の優先度のタスクの中でレディ・キューの先頭のタスクが選ばれています．

▶②CPUコアへのタスクの割り当て

実行するタスクを選択したら，そのタスクを実行するCPUコアを決めます．