

実行すべきタスクとその順番を決めるスケジューラ

豊山 祐一

第1章では簡易ディスパッチャを作成し、複数のC言語の関数をタスクとして並行実行させました。本章ではTry Kernelにスケジューラを実装し、OSの制御によりタスクが切り替えられるようにします。またタスクの生成、実行、終了といったタスク管理の基本的な操作を行うAPIも作成します。



スケジューラの作成

● スケジューラは実行するタスクを決定する

スケジューラは、実行するタスクの順番を決めるOS内のマルチタスク制御のプログラムです。

スケジューラは、スケジューリング規則に従って、実行できる状態のタスクの中から実際に実行するタスクを決定します。実行すべきタスクが、現在実行しているタスクと異なった場合は、ディスパッチャを呼び出して、タスクの切り替えを行います。

スケジューラが実行されるタイミングは、タスクの状態に変化があったときです。例えば、OSのAPIが実行され、新たにタスクが実行されたときや、実行中のタスクが待ち状態になったときなどが該当します。タスクの状態が変化すると、APIの処理プログラムの中でスケジューラが呼び出されます。

この他にスケジューラが呼び出されるタイミングとして、OSの時間管理処理や割り込み処理があります。

● スケジューリングの規則

Try Kernelはタスクのスケジューリングに、IEEE 2050-2018規格に基づいた優先度スケジューリングを採用します。スケジューリングは以下の規則で行います。

- 規則1：実行可能なタスクの中で最も優先度の高いタスクを実行する
- 規則2：規則1に該当するタスクが複数ある場合は、最初に実行可能な状態となったタスクを実行する

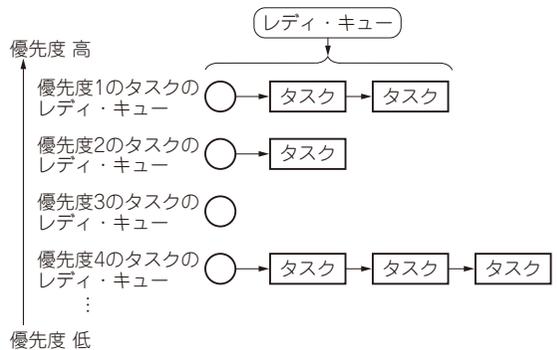


図1 タスクの実行待ちの行列をレディ・キューと呼ぶ

● タスクの待ち行列を作る

上記のスケジューリングを実現するために、優先度ごとにタスクの実行待ちの行列を作ります。この実行待ちの行列をレディ・キューと呼びます。

タスクが実行できる状態になると、そのタスクの優先度のレディ・キューの末尾につながれます。レディ・キューには実行できる状態になった順番にタスクが並ぶことになります。

レディ・キューの概略を図1に示します。図中では優先度1のタスクが2つ、優先度2のタスクが1つ、優先度4のタスクが3つ、実行できる状態で待ちになっています。優先度3のタスクはありません。この図では、優先度1のレディ・キューの先頭のタスクが実行すべきタスクです。

● 優先度スケジューリングの実現方法

スケジューラは以下のように処理を行うことにより、優先度スケジューリングを実現します。

- ①優先度の高い順番にレディ・キューを調べていく
- ②最初に見つかったレディ・キューの先頭のタスクを実行すべきタスクとする
- ③実行すべきタスクと現在実行中のタスクが異なればディスパッチャを実行する

レディ・キューに1つもタスクが存在しない、つまり実行すべきタスクがないという場合もあります。