

第1章 「同時に動く」ことによる効率化と設計の難しさ

並列処理の基本となるマルチタスク・プログラミングの基礎

五十嵐 仁

複数のタスクが同時に動作するマルチタスク OS は、複数の作業を同時に処理したり、リアルタイム性の要求される処理が容易に行えるなどのメリットがある。そのため、組み込み向け OS といえばマルチタスク OS といえるほど普及している。しかし、マルチタスク OS はシングル・タスク OS よりさまざまな要素を考慮しなければならない。資源の排他制御、優先順位、タスクの分割——これらの事項を把握して適切に設計する必要がある。そこで本稿では、並列コンピューティングの基本であるマルチタスク・プログラミングについて学んでいく。(編集部)

組み込みシステムの世界においても、マルチコア・プロセッサ対応の波が押し寄せてきています。その中でも、負分散型の SMP (Symmetric Multiprocessing ; 対称型マルチプロセッシング) に注目が集まり、SMP に対応した CPU や OS が続々と登場しています。

しかし、単純にシングル・コア環境で開発してきたソフトウェアを SMP 環境で動作させれば、性能が上がるわけではありません。それどころか正常に動作しない可能性すらあるのです(右掲の **コラム 1** を参照)。

マルチコア・プロセッサ環境では、複数の処理が完全に並列に実行されます。そのため、ソフトウェアを正常に動作させるには、適切な排他・同期制御を行う必要があります。と言っても、マルチコア対応のために、新しい特別な技術が必要となるわけではありません。基本となるのは「マルチタスク・プログラミング」です。

そこで、本稿ではマルチタスク・プログラミングの基礎について説明していきます。



図1 パソコンの環境では、マルチタスク・プログラミングは当たり前

1. マルチタスク・プログラミングとは

かつてのパソコン OS はシングル・タスクでした。同時に一つの処理しか行えず、複数の処理を行いたい場合は、一つの処理が終了したあとで、次の処理に取り掛かるしかありませんでした。

これに対してマルチタスクとは、1台のコンピュータ・システム(処理装置)で、同時に複数の処理(タスク)を並行して実行する機能のことです。例えば、1台のパソコンで CD の音楽を聴きながら、Web サイトを閲覧して、ワープロ・ソフトウェアでレポートを書くことができます(図 1)。このたとえでは、音楽再生ソフトウェアと Web ブラウザ、ワープロ・ソフトウェアを同時に実行させているので、マルチタスク環境を利用して実現していることになります。

近年の OS では、パソコン用だけでなく、組み込み用の OS でもマルチタスクが標準となっています。複数の処理を同時に行いたいという要求が高まっているためです。

それでは次に、マルチタスク・プログラミングがどのような原理で複数の処理を同時に実行しているかについて説明します。

マルチタスクを実現する方法

一番簡単にマルチタスクを実現する方法は、「タスクの数だけプロセッサを用意する」ことです。マルチタスクを実現したいコンピュータ・システムに、同時に実行するタスクの最大数以上のプロセッサ(コア)が搭載されていれば、空いているプロセッサに実行するタスクを割り振るだ

C O L U M N 1

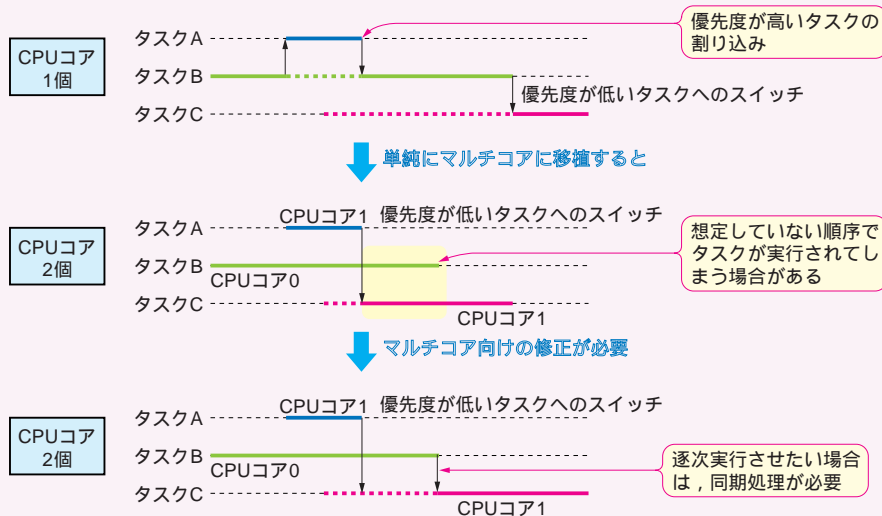
なぜシングル・コア・プロセッサのプログラムが SMP 環境で動作しないのか

シングル・コア環境で、やみくもに同期・排他制御の処理を実装した場合、逆に OS の処理がオーバヘッドになり、性能が低下する可能性があります。このような理由から、シングル・コア環境では同時に処理が実行されるタスクは必ず一つということを生かして、タスクの操作やオブジェクトを使わず、タスクの優先度に基づく実装を行っているのではないのでしょうか。

例えば、図 A のようにタスク A の結果を使用してタスク B が処理を実行するような場合、シングル・コアでは、単純にタスク A の優先度をタスク B より高くすることによって、タスク A が終了する前にタスク B が実行されないようにできます。一方マルチコアでは、タスク A がコア 1 で実行されているときにコア 2 が空いていると、タスク B より高い優先度の

タスクが存在しなければ、タスク B がコア 2 で実行されてしまいます。この場合、タスク A とタスク B の実行のタイミングを同期させれば、問題は解決します。例えば、タスク B は起動直後に起床待ちに遷移し、タスク A はタスク B が使用する結果の処理が終了した時点でタスク B を起床します。これだけで、正常に動作させることができます。

このように、シングル・コア環境のプログラムは、そのままでは SMP 環境で動作しません。しかし、ほとんどのケースでは、マルチタスク・プログラミングの基礎的な知識を持っていれば、SMP 環境へ対応できます。マルチコア・プロセッサへの対応の最初の一步は、マルチタスク・プログラミングを見直すことだと思います。



だけで、マルチタスク・プログラミングを実現することが可能です(図 2)。とは言っても、一般的なパソコンや組み込みシステムでは、経済性、消費電力、物理サイズの制約から、そのようなケースはほとんどありません。

そのため、マルチタスク・プログラミングでは、複数のタスクを高速に切り替えながら実行することにより、見かけ上、同時に動いているように見せています。具体

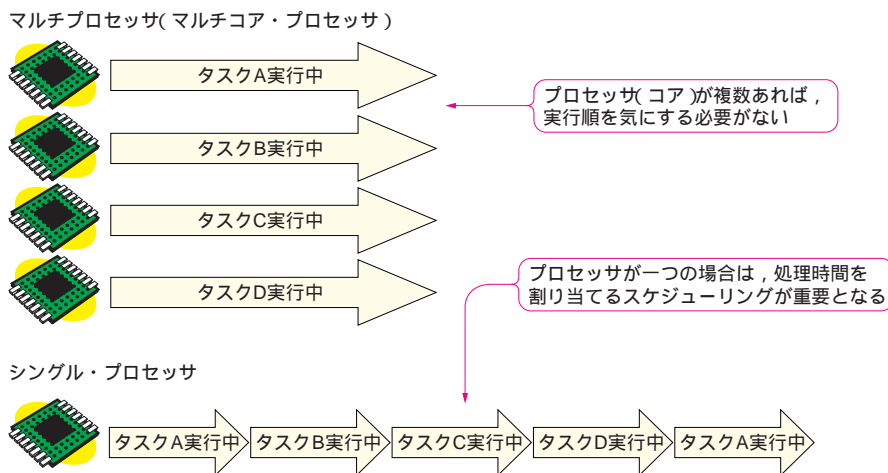


図2 プロセッサがたくさんあれば、マルチタスクは容易に実現できる