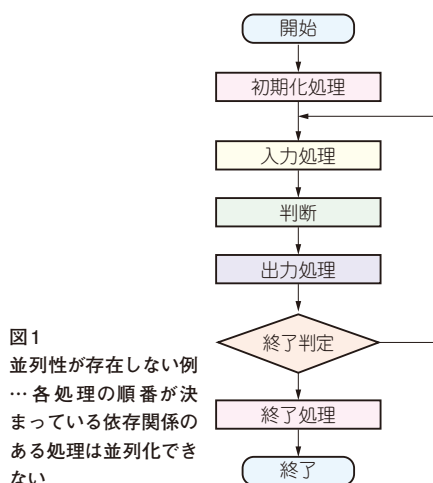


並列処理に向くプログラムの書き方ヒント

納富 昭



マルチコア・プロセッサ(以降、マルチコア)は消費電力を抑えつつソフトウェアの動作を高速化するためには必須の技術です。しかし、マルチコアをうまく使うソフトウェアの開発はなかなか難しいものです。本稿では、マルチコア上で動作するソフトウェアがどのようなものか、そのソフトウェアを開発するにはどうすればよいのかを最新のコンパイラ技術を交えて解説します。

データや処理の手順を並列化できるように、プログラミングするのが基本ですが、コンパイラによってはコンパイル時に一定の並列化を行える場合もあります。実際に処理を並列化し、画像処理のソフトウェアの動作を3.43倍に高速化する(4コアを使った場合)事例も紹介されています(コラム参照)。

本稿でもそのようなコンパイラで並列化できるかどうか試してみます。

マルチコアをうまく使える処理の条件

PCやラズベリー・パイなどのコンピュータで、マルチコアの性能を最大限に活用するには、そこで動作

するソフトウェアがマルチコアに対応している必要があります。マルチコアに対応していないソフトウェアでは、実行しても1つのコアしか使わないため高い性能は発揮できません。

では、マルチコア対応のソフトウェアとはどのようなものなのでしょうか。

マルチコア対応のソフトウェアを実現するには、

- 1) 実現対象の処理に並列性が存在すること
- 2) 並列性を生かせるようにソフトウェアをコーディングすること
- 3) 並列性を生かせる処理系(コンパイラなど)を使うこと

が必要です。

● 依存関係がある処理は並列化できない

例として、図1のような処理をソフトウェアとしてコーディングすることを考えてみます。

非常に大まかなフローチャートではありますが、「入力処理」で入力したデータを次段で「判断」する、という処理です。各段階の処理の出力が次段の入力となっており、これらの処理の順番は守らなければなりません。

このように処理の順番を守らなければならないことを、処理と処理の間に依存関係があると言います。依存関係があるのにそれを守らないと、正しい出力が得られません。

順番を守らなければならない処理が1列に長く続くようなソフトウェアには並列性が存在しないため、並列化できません。

● 処理に依存関係がなければ並列化できる

ここで、処理する対象のデータをもう少し詳細に分解したときに、図2のようになっていたとします。

入力は入力1、入力2に分解されます。入力1は出力Aを決めるための要素、入力2は出力Bを決めるための要素です。

このフローチャートの処理Aと、処理Bには依存関係がないため、並列に実行できます。