

第1章

C, C++, Fortranに追記するだけ!
対象は全てのマルチコアCPU搭載マシン

複数コアにおけるスレッド 並列化のための仕様 OpenMP

藤井 裕也

マルチスレッドのプログラムを作る場合、LinuxなどのOSやプログラミング言語が備えるマルチスレッドの仕組みを使ってスレッドを制御できます。

マルチスレッド・プログラミングでは同時に複数の処理が実行されるため、シングル・スレッドのプログラミングに比べて難しい場合が多いと思います。

ここでは、アセンブリ言語よりも扱いやすく、細かいスレッドの制御も可能なOpenMPという仕様を使って、GCCでマルチスレッド・プログラミングを行う方法を解説します。

ラズベリー・パイを対象に解説しますが、OpenMPは処理系に依存しないオープンな仕様なので、サーバやPCといった環境でも利用されています。

(編集部)

● 複数のコアを持つラズベリー・パイ

ラズベリー・パイは何度かモデル・チェンジしており、新しいモデルは、複数のCPUコア（以下コアと呼ぶ）を持っています。この複数のコアを同時に動かして計算をより短時間で実行することをスレッド並列と言います。

ラズベリー・パイのモデルとCPU、コア数の関係を表1に示します。より詳しい情報は公式サイト⁽¹⁾を参照してください。

▶ ラズベリー・パイ2には前期型と後期型がある

ラズベリー・パイ2が前期型か後期型かを調べるには次の方法が使えます。

- ボードの印字がRaspberry Pi 2 Model B V1.2なら後期型
- 次のコマンドの結果がRaspberry Pi 2 Model B Rev 1.2なら後期型

```
$ cat /proc/cpuinfo | grep Model
```

インテル製のプロセッサではコア数以上のスレッドを同時に実行できるものもありますが、ラズベリー・パイに搭載されているプロセッサは全て、同時に実行できる最大スレッド数とコア数は同じです。

● 複数コアを効率よく使うには

ラズベリー・パイに限らず、使用するコンピュータ

表1 ラズベリー・パイのモデルによるコア数

アーキテクチャ	コア数	対応するモデル
ARM1176JZF-S	1	ラズベリー・パイ Model A, B, B+, Compute Module, ラズベリー・パイ Zero
Cortex-A7	4	ラズベリー・パイ 2 Model B
Cortex-A53		ラズベリー・パイ 2 Model B V1.2, ラズベリー・パイ 3, Compute Module 3/3 +
Cortex-A72		ラズベリー・パイ 4 Model B

が複数のコアを持っていたとしても、普通にソースコードを書いただけでは、それらのコア全てを効率良く使用した計算がなされるとは限りません。

▶ マルチスレッド対応のライブラリを使う

複数のコアを生かして処理を行うためには、OpenBLASなどのマルチスレッドで動くライブラリを使うのが最も簡単です。データを事前に並列処理に適した形にしておくといった作業は必要ですが、プログラムが処理の並列性を気にしなくても複数のコアを有効に利用できます。

典型的な処理などは既にライブラリがあるので、それらの利用を検討した方がよいかもしれません。

▶ 自力でマルチスレッド・プログラミングする

欲しい処理に対してそのようなライブラリがない場合は、自分で記述するソースコードの中で、自力でスレッド制御をする必要があります。

スレッド制御の方法はプログラミング言語によってさまざまですが、C/C++の場合はOpenMPを使うのがお勧めです。

UNIX系OSで標準的なスレッドの作り方

OpenMPについて説明する前に、UNIX系のOSで標準的に使われるpthreadについて少し触れておきます。pthreadはPOSIX (Portable Operating System Interface, UNIX系OSが備えるべき共通の機能など) について、プログラムの呼び出し方などの標準を定め