



Object BLVD オブジェクトの散歩道

例題でわかる! .NET Framework

第5回 「スレッド事始め」

吉田 弘一郎 YOSHIDA, Koichiro



マルチタスク

~プロセスとスレッド

往年の、そして最近ではリニューアルされたらしい学園ドラマ「金八先生」の生徒たちは、授業中に勝手に言いたいことを言っているように見えますが、実はある種の“規則”を厳格に守っています。ドラマだから当たり前なのですが、生徒はみなさん“順にひとりずつ”発言しているのです。

ここで、「発言すること」をタスクと考えるなら、現実の教室では、あちこちで無駄話が同時進行するのが普通です。これは、コンピュータの中で、さまざまなタスクが同時進行するのに似ています。そうです、今月は「マルチタスク」の話です。

ファイルをダウンロードしている最中に、ワープロを使うことができるのも、マルチタスクだからです。いつのまにかマルチタスクという言葉は、あ

まりにも身近になりすぎてしまい、その有難さも感じないほどです。一昔前までは、マルチタスクの類は非常に高度な技法で、とても素人の手の届くものではありませんでした。実際、マルチタスクは相応のハードウェアとソフトウェアを必要としたのです。ところが、今日、文字通り、ありとあらゆるコンピュータが、悠々とマルチタスクをこなしています。パームトップの類にしても同様です。

マルチタスクを楽々とこなすPCは、みなさんの周りに無数にあります。Windowsもしかり、Linuxもしかりです。でも、マルチタスクのプログラミングの話になると、まだまだ敷居の高さを感じてしまいます。本当は非常に簡単なのですが、それほどポピュラーでないのも事実。その原因のひとつに、マルチタスク関係の用語法が確立していないということがあります。たいして標準化されていない標準規格がいくつもあります。ここでは、Windowsで標準と思われる用語を使いますが、それは次のようなものです。

「マルチタスク」とは、言うまでも

本稿で前提となるもの

OS Windows 2000 Professional (SP3) 以降

開発環境 Visual Studio.NET

.NET Framework 1.0.3705.288 (SP2)

Internet Explorer 6.0.2800.1106

初級

中級

上級

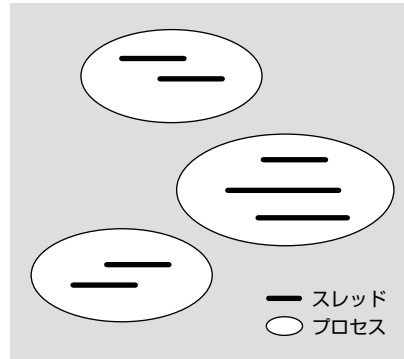
この記事で解説したサンプルプログラムは、付録CD-ROMの¥DMAG¥SAMPOフォルダ以下に収録しています。

¥SAMPO05¥SAMPO05.SLN : 今回紹介したサンプルのソリューションファイル
¥ANSWER¥VB01.TXT : 練習問題①の解答
¥ANSWER¥VB02.TXT : 練習問題②の解答
¥ANSWER¥VB03.TXT : 練習問題③の解答
¥ANSWER¥CS04.TXT : 練習問題④の解答
¥ANSWER¥VB04.TXT : 練習問題⑤の解答
¥ANSWER¥VB05.TXT : 練習問題⑥の解答

なく複数のタスクを同時にこなすことですが、「タスク」という概念が明確ではありません。まず、ここで、これを「プロセス」と「スレッド」に分類することにしましょう。各々の意味は以下のとおりです。

プロセス：通常の意味でのプログラム
スレッド：プロセス内部で走るサブプロセス

図1：プロセスとスレッドの関係



これは必ずしも目に見えるところで走るアプリケーションプログラムとは限りません。タスクマネージャの「プロセス」タブに実行中のプロセスがみな表示されますが、その多くは目に触れぬところでシステム関係の仕事を担当する「サービス」と呼ばれるものです。このようなサービスなどは決して普通の意味でのユーザーのプログラムではないのですが、「普通のシステム寄りのプログラム」であり、プロセスの一種なのです。このようなプロセスに対して、ユーザー寄りのワープロなどのプログラムは「アプリケーション」と呼ばれています。これもタスクマネージャの「アプリケーション」タブで見ることができます。

スレッドは、これら個々のプロセスの内部で同時走行可能な小さなプロセス（サブプロセス）のようなものです。このサブプロセスは“サブプログラム”とも呼ぶことができ、“サブルーチン”と混同されそうですが、実際、一種のサブルーチンのようなものでもありません。もっとも、サブルーチンは“呼び出す”のに比べ、スレッドの場合は“起動する”、ないしは“生成する”と言います。雰囲気的には次のような違

図2：タスクマネージャ



いです。

みなさんが「田中君」の上司であるとしましょう。この田中君がサブルーチンであれば、「田中君に仕事を頼んで、田中君がその仕事を完了するまで何もしないで待つ」ことになります。

一方、この田中君がスレッドであれば、田中君に仕事を頼んで、こっちはこっちで自分の仕事を続けることができます。さらに、山田君というスレッドもあるとしましょう。すると、「この山田君にも仕事を頼んで、こちらは独自に自分の仕事をし続ける」ことができるのです。

それでは、ここでWordを起動してみましょう。タスクマネージャを見ると、図3のように、プロセスがひとつ増えて“31”になりました。スレッドは“268”になりましたから、7つ増えたことになります。単純に考えると、Wordというプロセスが7つのスレッドをもっていることになります。実際、新しいスレッドの多くは新しいプロセスの中にあるのですが、そうでないスレッドがある可能性もあるので、ここでは深く追求しません。その代わりに後で簡単な例題を作り、これを実際に走らせてタスクマネージャで見てみることにし

図1のような感じです。

この図では、プロセスはみなスレッドを含んでいます。ここでみなさんは、「プロセスはみなスレッドをもっているのであろうか？」という疑問をもたれるかもしれません。この答えは、一般論としては“No”ですが、モダンな.NETの世界では“Yes”です。これに関しては、後で触れます。

プロセスとスレッドの具体例

例をあげてみましょう。みなさんのWindows PCで「タスクマネージャ」を起動すれば、「パフォーマンス」タブの左下に、図2にあるようにプロセスとスレッドの数が表示されます。この図は私が目下原稿を書きつつある日本語版Windows 2000での例ですが、261のスレッドと30のプロセスが実行中であることを意味します。ここで、“実行中”と言いましたが、正確には「起動された状態にある」と言うべきでしょう。起動されたが、何もやることがないので仮眠中のプロセスもあるでしょうから。

さて、ここで「プロセスとは通常の意味でのプログラム」と書きましたが、