

Interface誌2007年5月号に付属したV850マイコン基板で動作するオリジナルOS「MicrOS」が登場した。今回は、MicrOSにおいてリアルタイム性を確保するための割り込み処理部分について見ていく。（編集部）

1. リアルタイム・システムとは何か

● リアルタイムであるということ

組み込みシステムの分野では、「リアルタイム OS」(RTOS, Realtime Operating System)という言葉が一般的に使われています。そして、リアルタイム OS を使ってさまざまなシステムが作られています。

しかし、ここで間違えてはいけないのは「リアルタイム OS だからといって、必ずしもリアルタイム・システムを制御できる OS ではない」ということです。また、逆にリアルタイム OS といわれない OS はリアルタイム・システムを制御できないかということ、そうでもないのです。

クリティカルな機能を決められた時間内に処理するように構成されたシステム、というのがリアルタイム・システムの一般的な定義です。この時間的条件を必ず満足しなければならないシステムをハード・リアルタイム・システム、もし守れなくても致命的にはならないようなシステムはソフト・リアルタイム・システムといわれます。

● 時間的な制約が存在する

問題は、この時間的な制約です。この点に関しては次のように解釈すべきでしょう。

例えば、相手が人間である場合は、数秒間以上応答がないとイライラするかもしれません。もし、制御対象が携帯電話やPHSのような通信機器であったとすると、通話中は数ms単位でパケットのやり取りが行われるので、これに間に合うような処理をしなければなりません。UART経由で制御する機器があったとすれば、ボーレートが関係します。9,600bpsであれば1バイトの転送は約1msなので、やりとりされるコマンドの処理がこの条件を満たしていなければならないことになります。USB制御の機器であれば

その性能を引き出すために、もっと厳しい条件での処理が要求されます。

例えば、DMA (Direct Memory Access) のような機構をUARTに適用すれば、1文字ごとに発生する割り込みを「一塊のデータ単位ごと」にすることができます。厳しい条件が要求される機器については、専用のハードウェアによってある処理の一部を代行してくれるので、リアルタイム制御が可能となります。

このように時間的制約は、制御の対象となる相手(機器)によって異なります。それも組み込み機器の動作が速ければ速いほど、より高い応答性能が要求されます。

非常に高速な応答が要求される場合、OSではなくハードウェアの側で問題を解決した方がよいときもあります。そのような場合、OSは専用化され、場合によってはCPU自身も専用化されます。

2. リアルタイム・システムを制御するOS

それでは、リアルタイム OS とはいかなる OS をいうのでしょうか。リアルタイム性が優れているといわれるRTOS「 μ ITRON」で、リアルタイムのための機構とはどのようなものか調べてみました。

● 優先度ベース・スケジューリング

優先度ベース・スケジューリングとは、常に優先度の高いタスクが実行されるように制御することです(図1)。同じ優先度であればFCFS (First Come First Service) で制御します。

● プリエンプティブ・スケジューリング

プリエンプティブ・スケジューリングとは、あるタスクが実行中にそれより高いタスクが実行可能状態になったと

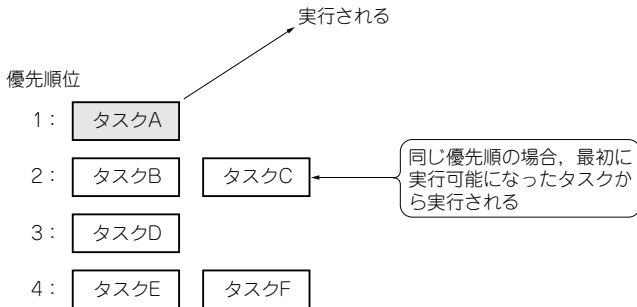


図1 優先度ベース・スケジューリング

優先順位が1~4のタスクA~Fがあった場合、一番優先順位の高いタスクAが実行される。

タスクAの次は優先順位2のタスクBかタスクCが実行されるが、タスクBが最初に実行可能になったので、タスクB→タスクCの順に実行される。

きに優先度の高いタスクに切り替えて実行することです(図2)。この動作のきっかけになるのは、割り込みの発生です(内部割り込みと外部割り込みのあるタスクが別のタスクを起動する場合、内部割り込みを使っていることが前提)。

この二つのスケジューリングは、μITRON系のリアルタイムOSで採用されています。FCFSの代わりにラウンドロビン方式を採用しているリアルタイムOSもあります。ラウンドロビンとは、同じ優先度のタスクに対してはできるだけ並行して処理を進めていくように制御する方式です。

● MicrOSはFCFSとラウンドロビンの両方に対応

MicrOSはFCFSとラウンドロビンのどちらのスケジューリング方式でも選択できるようになっています。これを切り替えるのは簡単です。実行していたタスクをレディ状態に戻し、さらにプライオリティ・テーブルに戻すときにリンク・セルの先頭に戻すか最後尾に戻すかという違いだけです。このようなタイミングは、外部割り込み処理とタスクのウェイトを解除するようなシステム・コールが発行されたときだけです。いずれも__active関数をコールして処理しています。このなかでMicrOSの定義変数によって処理方法を切り替えることにより対処しています。

● リアルタイム性に関する議論

リアルタイムOSの定義がタスク・スケジューリングの部分だけであれば、一見、UNIX系OSのリアルタイムOS化はそれほど難しくないように思えます。しかし議論が沸騰しているところを見ると、OSのリアルタイム性というのはそれほど単純でないのかもしれませんが。

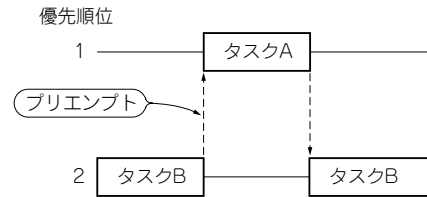


図2 プリエンプティブ・スケジューリング

最初はタスクBが実行されていたが、ある瞬間にタスクAも実行可能になったとする。その場合、タスクAの方がタスクBよりも優先順位が高いので、タスクBの実行を中断し、タスクAを実行する。このように、実行権を横取りすることをプリエンプト(プリエンプション)と呼ぶ。

「リアルタイム」は「時間的に処理を間に合わせる(時間通りに処理する)」という意味でした。しかし、特定のシステムを対象としていないOSに対して「リアルタイム」という形容を行うのは、無理があるのかもしれませんが。やはり、「リアルタイム・システムを制御することができるOS」程度に理解しておく方が安全なのかもしれません。

リアルタイム性が足りないと表現されたとき、それは単に処理が間に合っていないことが多く、システムのチューニングなどによって解決する場合もあるからです。

● 現実的な「リアルタイム性」の解釈

しかし、特定の組み込みOSのリアルタイム性について説明を求められたとき、上述のようなあいまいな表現は許されません。そんなときの答え方の一つをここに示しておきましょう。

リアルタイム性を、タスクが発生し、そのタスク処理を行い、結果を返すまでの時間と仮定します。これをOSの処理に当てはめれば、「割り込みが発生して、それを受け付け、割り込みを処理して、次の処理に移行できるまでの時間」と考えられます。この時間がある一定の時間内に収まっていれば、そのアプリケーション・システムのリアルタイム性が保証できることになります。

● MicrOSでのリアルタイム処理部分

各部分の処理はどの程度のステップ数になるのでしょうか。外部割り込み処理とディスパッチャの処理をMicrOSの場合で詳細に示します。

外部割り込み処理の流れをリスト1に示します(理想的なMicrOSの場合)。次に、ディスパッチャの処理をリスト2に示します。

外部割り込み処理は多重割り込み処理を行うことを前提に、その処理を示しています。このときV850を前提にす