

南角先生の組み込み講座

南角 茂樹
大阪電気通信大学 准教授



第2回 割り込み

今回は RTOS とも密接に関係し、RTOS を実行させるのに欠くことのできない割り込みの関係の話をしてしたいと思います。

前回、現実世界の変化にリアルタイム性を保ちつつ対応できることが RTOS の目的であるという話をしました。ではどうやって組み込みシステム CPU は外部環境の変化を知ることが出来るのでしょうか？

具体的には次の3種類の方法があります。

- 割り込みを使用する
- ポーリングを使用する
- ポーリングの特殊型でもある、スピンロックを使用する。

通常は割り込みとポーリングという分類ですが、今回はさらにスピンロックを加えました。

最初に上記の3つを日常生活でのアナロジーで説明します。

仮に現在9時で、10時になったらあるところに電話をしなければならないとします。そのような時皆さんはどうしますか？

- 一番多いのはアラームを10時にセットしておいて、他の仕事をするด้วยですね。これが割り込みに当たります。普通に考えるとこれが一番良いようですが、アラームが鳴った瞬間にやっていた仕事を、電話の後で再開するための情報を記録しておくという作業が必要になるかもしれません。その記録を取る時間だけ電話をかけるのが遅れることとなります。さらにアラームをセットする作業も必要となります。これが割り込みに当たります。
- 次に多いのが、ほかの作業をしながら、キリがいいところで定期的に時計をチェックすることです。
この場合はほかの仕事をしていても、何らかの適切な切れ目切れ目で時計をチェックする訳ですから、再開するための情報の記録は必要ないか少ないことになり、時計をチェックしてから電話をかけるまでの時間はアラームに比べると少ないということになります。ただしお気づきだと思いますが、時計を見るタイミングによっては、電話をかける時刻に遅れる場合もあります。また時間前であってもたびたび時計をチェックしなければならないという無駄もあります。これがポーリングにあたります。

- 最後の方法は、現実世界では通常は行いませんが、残り時間が1分など短い場合にはかなり一般的な方法です。それは時計を見続けていてほかの作業はしないというやり方です。ずっと時計を見続けていてほかの作業はしないで、時間になると同時に電話する訳です。この場合は再開しなければならない仕事はないため、時間にも正確です。もちろん他に有効な仕事をできないという欠点があります。これがスピンロックあるいはビジーウェイトにあたります。

あとで具体的に話しますが、この一見古めかしくて無駄が多いと思われるスピンロックが、コンピュータの世界ではかなり重要な技術でもあり、実はかなり使用されています。たとえば別のアーキテクチャのCPU間での排他制御、マルチコア対応のLinuxでコア間の排他制御などにはこのスピンロックが使われます。これに関しては次回以降にお話ししたいと思います。

以上で大まかにイメージをつかんでいただいたと思うので、次にアナロジーでなく具体的に説明します。最初に割り込みを説明します。実は割り込みといっても、表1に示すようにいくつか種類があります。停電など、通常はシステムにとって致命的な状態に対応するため禁止することができない割り込みであるNMI (Non Maskable Interrupt)、TLBミスやなど未定義命令などソフトウェアの実行によりCPU自体が発生する割り込み、RTOSによってはアプリケーションからのOS呼び出しであるSVCにも使用する(場合がある)ソフトウェア割り込み、そして外部(ハードウェア)割り込みなどです。以後はRTOSに関係が深く、またデバイスドライバとも密接に関連する外部割り込みを対象を絞ります。

種類	発生要因
センサー/アクチュエーターなどの入出力装置(デバイス)からのリクエスト	外部
仮想メモリ(TLBエラー)	内部
ユーザープログラムからのOSの起動(CPUモードの切り替えを含む時)	内部
オーバーフロー	内部
未定義命令の使用	内部

表1 割り込み

割り込みとは、プログラム実行中に発生する何らかのイベント(事象)により、プログラムの実行を一時中断して、他のプログラムに制御を移す機構で制御が移されたプログラムが実行を終了すると、元のプログラムを再開させます。プロセッサは、プログラムの各命令の実行毎に割り込みイベントの発生をしらべ、実行中のプログラムの優先度より高いイベントが発生している時に、このイベントに対応するプログラムに制御を移します。この割り込みイベントにより実行されるプログラムは割り込みサービスルーチン(ISR)や割り込みハンドラと呼ばれます。

たとえばセンサなどの周辺デバイスで検知されたイベントに対してリアルタイムに応答する必要がある場合は、そのイベントを知らせる周辺デバイスの信号線をマイクロプロセッサの割り込み信号ピンに接続して、その端子に変化があった場合に、その変化をトリガとして、CPU自体が現在実行中の処理を後で再開させるためのデータを保存した後、予め指定したプログラムの実行を開始させる機能です。特にRTOSを使用しない場合にはプログラムを並行実行させるための重要な機能ですし、次回お話しするように実行中のタスクを制御する

ために周期的に RTOS に制御を切り替えるにも割り込みを使用します。割り込み発生時には CPU 自身が自動的に以降の割り込みを禁止しますが、ソフトウェアでも割り込みを禁止したり許可したりもできます。排他制御においても割り込み禁止/許可は重要な技術で、重要な処理をする前にあらかじめ割り込みを禁止しておいてから処理を開始する場合があります。先のアラームの例でいうと重要な仕事をする場合には、電話をかけるのが遅れることも覚悟して、あえてアラームを停止する場合もあるかも知れません。それが割り込み禁止です。

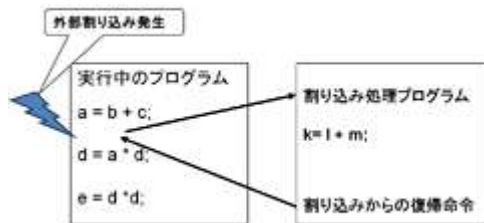


図 1 割り込み発生時の動作

図 1 は割り込み発生時の動きを示す図です。プログラムのある命令を実行中に割り込みが発生した場合、その命令の実行終了時に CPU ごとに決められている（あるいは決めることができる）あるアドレスにある命令の実行に制御を移します。ソフトウェアでプログラムの流れを変える場合はジャンプ命令やサブルーチン呼び出し命令（ARM では PC レジスタに対するアドレスのコピーですが）、サブルーチンや ISR からの復帰命令が主な手段ですが、それに対して（外部）割り込みは呼び出し元のソフトウェアは存在しません。いわばハードウェアによるソフトウェアの呼び出しです。割り込みから、割り込まれたプログラムにそのまま復帰する場合は、サブルーチンからの復帰命令に似た役割の専用の割り込みからの復帰命令を使用します。割り込みからの復帰の場合は実行するプログラムのアドレスの復帰以外に、CPU の状態の復帰という機能も必要だからです。具体的には ISR 終了時に再開させるプログラムのアドレス（一般的にプログラムカウンタと呼ばれるもの）とプロセッサの状態を保持している特殊レジスタ（一般的には PSW？プロセッサステータスワードと呼ばれるもの）を復元します。

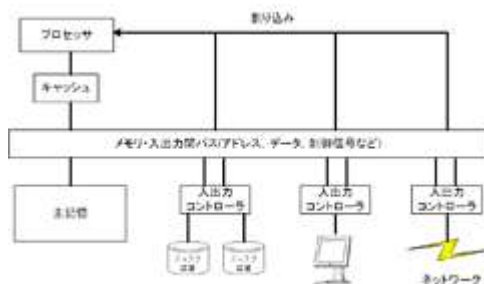


図 2 割り込みの利用方法

図 2 はコンピュータシステムにおける割り込みの利用法を示します。入出力コントローラというのは組込みシステムにおいては各種デバイス、ペリフェラルを示しています。情報処理の分野では伝統的にチャンネルと呼ばれているものです。

割り込み発生時 CPU が自動的に割り込みを禁止してから ISR を呼び出すという話はしましたが、RTOS 使用時は割り込みに関しは ISR の中でソフトウェアにより割り込み許可を行うのが一般的です。これを多重割り込み許可といいます。多重割り込みを許可した場合の複数の ISR の動きを図 3 に示します。これはタスクの動きとも似ていますが、タスクとちがい ISR の呼び出しの関係は入れ子になっていなければならないなどの違いもあります。

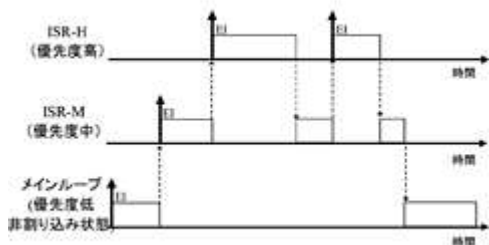


図 3 多重割り込み

今回は RTOS と割り込みの関係をお話ししたいと思います。

第 2 回おわり

コラム① 割り込みの優先度

CPU によっては、割り込みにも優先度があるものとなないものがあり、RISC 系 CPU には割り込みに優先度がないものが多いです。リアルタイム性を守るためには、あらかじめ動作の順序が想定できる優先度つきのほうが設計しやすいです。そこで RISC で組込みシステムを構成する場合には、RISC の備える割り込みの種類が少ないこともあり、外部に優先度付きで割り込みを処理できるデバイスである PIC (Programmable Interrupt Controller) を付加したり、VxWorks のようにハードウェアが優先度付きの割り込みをサポートしていない場合は、ソフトウェア的に割り込みベクタテーブルを実現することにより、アプリケーションソフトウェアの変更を最小限に減らす工夫をしたりしています。そのため本連載では割り込みに優先度があるものとして話をすすめます。

コラム② 昔から組込みシステムではマルチプロセッサは当たり前

会社に勤めていた時代から親しくしている某企業のハードウェアエンジニアである T 君は、10 年以上前から組込みシステムがマルチプロセッサなのは当たり前だとずっと言っていました。彼が言うのはもっともで、いわゆる情報処理の世界ではチャンネル、組込みの世界ではペリフェラル、デバイス、コントローラ、周辺機器、周辺デバイス、I/O などと呼ばれるものもある意味プロセッサです。たとえばノートパソコンであれば、現在であれば Corel-7 や 5 などの X86 系の CPU が 1 つに対して、ハードディスクコントローラ、ネットワークコントローラ、LCD コントローラ、キーボードコントローラなど多くのデバイスで構成されています。これらのデバイスは、CPU などほかのデバイスの要求に対してある動きを行う、機能特定の一種のプロセッサといえます。そうであれば、組込みシステムの大部分は、非対称マルチプロセッサシステムと呼んでよいわけです。私自身も、状態を持ったデバイスはプロセッサとよんでもいいと思いますし、T 君の意見になるほど思っていました。