

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



【組み込みシステム編】第6回 割り込みまとめ

こんにちは、南角です。

先日 NHK BS プレミアムで「火星 大冒険！～地球外生命を発見せよ～ 8月 NASAの探査機キュリオシティが火星に着陸」が放送されました。

出演者の方などは、もちろん探査機キュリオシティが何をできるのかとか、その着陸の仕組み、火星人はいるのかなどに興味津々でしたが、私の興味の中心はもちろん、組み込みシステムとしてのキュリオシティ自体です。この連載を読んでくださっている方の多くの方も同じではないかと思います。ご存知だとは思いますが、キュリオシティにも VxWorks が使用されています。

ところで、宇宙空間では放射線などの影響でメモリのビット化けなども起こりやすいため、かつては、一部の宇宙機器では定期的リセットを入れるなどの方式が使われてきました。

リセット解除後の動作は、決まった処理から実行され、基本的には初期化から実行されるためです。

実は、展示会でパチンコの組み込みシステムでも、同じように定期的リセットを入れる方式もあると聞いて面白く思いました。

パチンコ台の裏側も、ノイズが激しく、プログラム（CPU）が暴走しやすいためだそうですが、宇宙機器とパチンコという、一見何の関連もないような組み込みシステムの二つの分野で、似たような技術が使われていることが興味深かったです。

ところで、リセット信号も割り込みです。

意外と CPU の割り込み処理について勘違いされている方がたがおられるので、この連載の最後に割り込み処理について、その処理方式を中心にまとめておきたいと思います。

割り込みにもソフトウェア割り込みや、プログラムのエラーに起因する割り込み、マスク不能割り込み(NMI)などもありますが、ここでは外部（ハードウェア）割り込みに話を絞ります。

組み込みシステムの大きな特徴が、現実世界、外部環境の変化に応じて、制約時間以内に処理結果を出す(リアルタイム性の実現)があります。そして現実世界の変化を知るための主な手段が、各種センサからの割り込みです。

関数が他のソフトウェアから呼び出される処理なのに対して、割り込みハンドラや割り込みサービスルーチン(以後 ISR)はいわば、ハードウェアから呼び出される処理とも言えます。

ただし、関数が単に実行する命令のアドレスを変更するだけなのに対して、割り込みはそれ以外に、CPUの状態を変化させます。そのため CPU ハードウェアは戻りアドレスだけでなく、CPUの状態を保持しているプロセッサステータスワードなども保存します。

ところで現実世界の変化に対応して処理を行う組込みシステムですが、現実世界の変化は同時多発的に発生するため、ある ISR 実行中により優先度の高い割り込みが発生した場合は、その ISR に制御を移すという多重割り込みを利用して、多重処理で対応しています。

この時、ソフトウェアで処理を行わない場合は、外部割り込みが発生すると CPU ハードウェアが以後のすべての外部割り込みを自動的に禁止するため、多重割り込みが行われず、たとえ CPU ハードウェアが優先度付きの割り込み機能を備えていたとしても、多重処理は実現できません。割り込み発生時に呼び出された ISR 内で割り込み許可命令を実行することにより、多重割り込みの動作を行います。

CPU ハードウェアが割り込みを自動的に禁止するという動作の必要性は、RISC 系の CPU のように割り込み発生時に戻りアドレスや PSW を保存しておく場所がスタックではなく、専用レジスタの場合を考えれば、理解できると思います。

割り込みスタックのように、割り込みが発生するたびに保管場所が自動的に変わる場合であっても、外付けデバイスの設定保存など、次の割り込みが発生するまでに実行しておかなければならない処理が存在する場合も多く、そもそも一番最初に発生するリセット割り込みでは、スタックポインタの初期設定など各種初期設定を済ませる前に次の割り込みが発生するとまずいですよね（その意味では、リセット割り込みは戻りアドレスの保存が行われないなど、特殊な割り込みではありますが）。ハードウェアがすべての割り込みを禁止するという動作は、必要だと思います。

CPU ハードウェアが割り込みに優先度を備えている場合は、ISR 内で割り込み許可命令を実行した後は、実行中の ISR を呼び出した割り込みよりも、より優先度の高い割り込みが発生すれば、それに対応する ISR に制御が移るという多重処理になります。

なお、割り込みに優先度がない RISC 系の CPU の場合は、ソフトウェアで割り込みに優先度をつけたり、あるいはそもそも RISC 系の場合は割り込みの数そのものが不足するため、外部に割り込み優先度を備える PIC(Programmable Interrupt Controller)を付加することも多いため、以後割り込みは優先度付きの多重割り込みとして話を進めます。

RTOS が提供する機能であるタスクを用いて並行処理を実現する場合、割り込みに対する応答性を確保するため、割り込み処理自体での処理は最小限にして、処理の大部分をタスクで実行する方式がよく使われます。この時も通常は多重割り込みを許可しているため、排他制御が必要な処理は ISR 中では実行しない、あるいは、RTOS が排他制御が必要な場合は CPU ハードウェアが備える、すべての割り込みを禁止する(シングル CPU の場合)、などの手段を用いて、排他制御を実現します。

比較的良好に用いられるのが、単独のタスクによる割り込み処理の実行です。割り込み処理から依頼されて必要な処理を行うタスクを一つにまとめたものです。

この方式を図に示します。

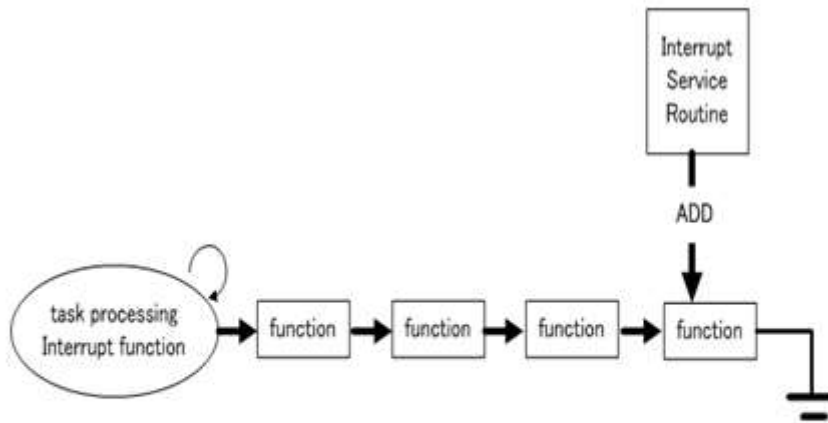


図1. 割り込み処理を一つのタスクで行う場合

割り込み発生時に呼び出される ISR 中では、必要最低限の処理のみを行い、その他の処理は、依頼された処理を実行するための、専用の優先度の高いタスクに依頼します。ISR はそのタスクに依頼内容を伝えるための処理要求キューに処理を追加し、タスクは処理要求キューにつながれた処理を順次実施します。

割り込み処理をタスクに依頼するための処理要求キューは FIFO 方式であるため、割り込みの優先度が処理の優先度に反映されないという問題もあります。

ハードウェアの割り込み優先度に応じて、このキューを優先度付きのキューにすると、キューにつながれた処理間での排他制御により、デッドロックを引き起こす可能性が高いため、このキューを優先度付きのキューにするのは難しいためです。似た方式で優先度の高い割り込み処理からの要求を、タスクではなく優先度の低い周期的なタイマー割り込みによる周期起動 ISR で処理する方式もありますが、専用タスクで処理する方式と同様に、割り込みからの要求を保存するキューが FIFO 方式で実装されているため、処理が割り込みの優先度に対応できないという問題があります。

その他にも、割り込み機能を単純に利用した方式もあります。これは ISR 自体の実行はハードウェアが実行する機能をそのまま使用した、つまり多重割り込みを禁止した状態にしておき、その代り ISR が行う処理は、処理が必要かどうかを示すフラグを ON にするだけにしたものです。そして、非割り込み環境で実行しているメインループが無限ループとなっていて、フラグに応じて必要な処理を行う方式です。この方式は実質的な処理を行うのがメインループのみであるため、排他制御の問題は発生しません。

しかし逆にそのことにより、処理に優先度をつけることができなくなり、前に説明した 2 つの方式と実質的に同じ問題があります。

そのほかにも ISR を扱う方法はいくつかありますが、それを調べる、あるいは新しい方式を考え出すことは皆さんの今後の課題としておきます。

ISR 間の排他制御には、タスク間の排他制御のようにセマフォというものを使用することはできません。

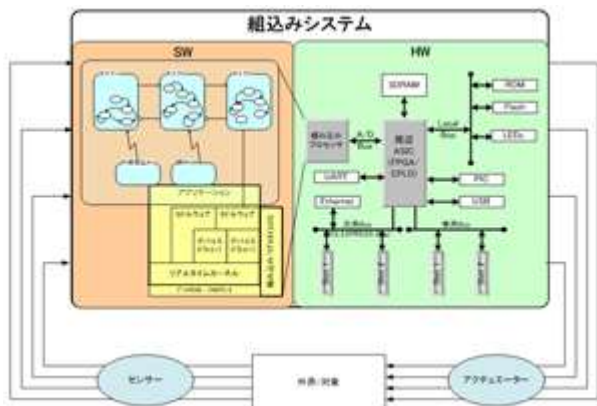
RTOS 自体もその多くの処理は、周期的なタイマー割り込みを使用する、割り込み処理です。

RTOS 用のタイマー割り込みの優先度をどのレベルにするのか、RTOS は ISR との排他制御をどのように実

現しているのかなどの課題があります。

このあたりの理解を深めていけば、RTOS を使用する場合でも、RTOS をブラックボックスとして扱わなくてもよくなってきます。現在でも、RTOS を使用することはごく当たり前になっていますが、RTOS をブラックボックスとして使用すること、その原理や構成を知って使いこなすのでは、出来上がったシステムの完成度に大きな差があるということを述べて、最終回とします。

最後に、前にも記載しましたが、組み込みシステムの表現する図として私が好きな図を再度記載します。



この連載も最初の予定の半年よりもかなり長くなりました。ありがとうございました。今後も何かの機会にお目にかかると思いますが、よろしくお願いします。