

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3回 RTOS における ISR の実行

こんにちは、南角です。

今回はすこし内容を変えて、実行すべき処理がない時のスケジューラに関して考えてみたいと思います。

20年以上前に、ある組み込みシステムに関するメーリングリスト上で議論したことがある話題です。

RTOS 使用時、最初に実行するのは割り込み処理です。実行すべき割り込み処理がない場合に、初めて RTOS のスケジューラは実行可能（レディー状態の）タスクを実行させるのですが、実行可能状態のタスクも存在しない場合の、スケジューラの動作です。

この時の RTOS 動作をどうするかで、RTOS のスケジューラのつくりが変わってきます。

スケジューラの構造を単純にするためには、そのような状態を無くせばよいことになります。

つまり、実行可能なタスクがひとつもないという状態にならないようにすればよいことになります。

この実現方法は、実は簡単で、優先度が最低のタスクで、処理を終えることがない処理（よく無限ループと呼ばれる、C では `for(;;){...;}` と書かれることの多い処理です）を実行するだけです。

このタスクを idle タスクと呼ぶことも多いです。

この場合は、逆に idle タスクが実行している時は、システムで本来実行すべき処理がないということなので、システムとしての余裕時間とも考えられます。

idle タスクは常に実行可能なわけですから、これを設けることにより、システム内に実行可能なタスクがないという状態は存在しないことになり、スケジューラにその場合の特殊処理を設ける必要がなくなり、スケジューラの構造を単純化できます。

この構造の問題点は、常に実行する処理があるということです。

CPU のアーキテクチャにもよりますが、CPU によっては、例えばソフトウェアで実行できる STOP という命令を備えていて、その命令で大部分の CPU の動作を止めることができます。

STOP 状態であっても、割り込みが発生すれば、再び元の状態に復帰して割り込み処理の実行を再開できるため、動作としては問題ありません。

この方式の利点は、省電力が可能なことです。

尤も CPU の動作クロックを、ソフトウェアにより下げることが可能な CPU であれば、idle タスクの中で

CPU の動作クロックを下げることにより、省電力は可能になります。

そのような CPU であっても、割り込みが発生すれば、動作クロックはハードウェアでもとに戻るため、動作に問題はなりません。

一方、実行しなければならない処理がない時の動作を RTOS 自身で実現している場合の、スケジューラの処理の概要は次のようになります。

```
if (実行すべき割り込み処理が存在する) {
    割り込み処理を実行する;
}
else if (実行可能なタスクが存在する) {
    実行可能なタスクを実行する;
}
else {
    CPUの動作を一時停止する;
}
```

CPUの動作を一時停止させている場合でも、システム割り込みなど何らかの割り込みが発生すれば、CPUの動作はハードウェア的に正常にもどり、再びスケジューラが呼び出されるわけですから、システムの動作には問題はありません。

復習ですが、上記の処理実行中に割り込みが発生した場合、割り込みから戻る時は、割り込まれた場所に直接復帰します。上記の”実行可能なタスクを実行する”によりタスクが実行されている場合の割り込みのみ、コンテキストを保存後、このスケジューラの先頭に制御が移ります。

みなさんがスケジューラを設計するとしたら、どちらの構造にしますか？

なお、上記のスケジューラの処理は、割り込み処理に関しても、すべてスケジューラで実行するものとして記述しており、VxWorksのスケジューラとは構造が異なっていることをお断りしておきます。

今回は、図もコラムもないシンプルな形にまとめてみました。

物足りない方のために、最後に宿題を出しておきます。

以下の2種類の関数をC言語で作ってみてください。C言語なので参照は使用不可とします。

一つ目は、ありふれていますが、2つの整数を入れ替える関数 Myswap、プロトタイプは次の通りです。

```
void Myswap(適切な引数);
```

もう一つは3つの整数の整数値のうち、真ん中のものを戻り値として返す関数 Mymid です。

プロトタイプは次の通りです。

```
int Mymid(int, int, int);
```

この連載を読んでいるかたならば、ただこれらの関数を作るだけであれば、簡単なことだと思います。そこで、効率や実行速度とは全く別の観点で、人とは違う面白いプログラムを作ってみてくださいという条件を加えます。組込みらしく両方ともビット演算を使用してみてください。

では、また次回。