

自動車制御用リアルタイム OS 仕様 (案) に関する検討結果

RTOS 自動車応用技術委員会

編: 高田 広章 (豊橋技術科学大学 情報工学系 / ITRON 専門委員会)

31-Mar-1998

このドキュメントでは、トヨタ自動車(株) 第3電子技術部からの「 μ ITRON3.0 仕様サブセット: 自動車制御用リアルタイム OS 仕様(案)」の提案を受けて、RTOS 自動車応用技術委員会で検討を行った結果について述べる。トヨタ自動車からの提案(これを原提案と呼ぶ)に本ドキュメントのコメントを加えたものを、 μ ITRON4.0 仕様の1つのプロファイルにすることを、RTOS 自動車応用技術委員会から提案する。なお、本ドキュメントに残っている未確定部分については、 μ ITRON4.0 仕様の検討の中で決定するものとする。

1 SUSPEND 状態の必要性

原提案にもある通り、ソフトウェアによるデバッグ時には SUSPEND 状態は有益であるが、製品組込み時には必要のない機能と考えられる。システムの信頼性を保ったまま、デバッグ時にのみ SUSPEND 状態をサポートするコードを組込み、製品からは外せるのが理想的である。

2 コンテキスト退避領域の共有化について

コンテキスト退避領域の共有化は、RAM を節約するためには非常に効果的であり、ぜひ導入すべきである。ただし、原提案では同一優先度のタスクはすべてスタックを共有するとしているが、他のタスクと同じ優先度で待ち状態を使いたいタスクもあると考えられるので、同一優先度のタスクの中で指定したものだけがスタックを共有するように一般化すべきである。スタックを共有していないタスクは、待ち状態に入ることができる(方式1)。

コンテキスト退避領域を共有化するもう1つのアプローチとして、「待ち状態のない μ ITRON カーネル仕様案」と整合させる方法がある。具体的には、「待ち状態のない μ ITRON カーネル仕様案」では、タスクが同一優先度でない場合にも、(待ち状態に入らないために)同じスタック領域を共有することができる。そこで、待ち状態に入れられないすべてのタスクは1つのスタックエリアを共有し、待ち状態に入れるタスクには個々にスタックエリアを割り付ける方法が考えられる(方式2)。

この2方式の長所・短所は次のように整理できる。

1. 方式1の長所・短所

- スタックを共有しているタスクは、
 - 待ち状態に入ることができない。
 - 優先度を初期優先度以下に変更してはならない。
 - 他タスクから優先度を変更してはならない。
 - 途中で終了することができる。
- 同一優先度のタスクでのスタックエリアの共有は、RAM の大幅な節約に直接つながる。
- 実現が容易である。
- μ ITRON 仕様に対して考え方の変更が少ない。

2. 方式2の長所・短所

- 待ちに入れられないタスクは、
 - 待ち状態に入ることができない(当然)。
 - 優先度を初期優先度以下に変更してはならない。
 - 他タスクから優先度を変更してはならない。
 - 途中で終了することができない(この制約は外すことも可能)。
- 待ちに入れられないタスクのコンテキスト保存領域を小さく押さえることができるため、RAM の節約につながる。
- 待ち状態のない μ ITRON カーネル仕様の考え方と統合するのに都合がよい。
- OSEK/VDX OS の考え方と類似している。

以上より、この2方式には一長一短があるため、どちらが適切か慎重に検討する必要がある。

3 キューを用いたタスク間通信

原提案では、メッセージのキューとデータのキューが別々に定義されているが、システムコールに渡すパラメータのデータ型以外は同じ機能を持つものである。そのため、カーネル内には片方の機能を持てば十分である。アプリケーションでは C 言語の型チェック機能

を活かしたいという要望に対しては、インラインを定義することで同等のことを実現できる。

キューが一杯になった場合に、エラーで戻る方式と、最も古いデータを上書きする方式の両方のサポートが望まれる。

4 個別のシステムコールについて

4.1 get_tid

原提案ではサポートしないこととなっているが、特に取り除く必要はないと考えられるので、プロファイルに含めるべきである。

4.2 タイムアウト

タイムアウト付きのシステムコールをサポートするとカーネルが大きくなるのは確かであるが、外部からのイベントやメッセージを待つような場合には有益である。カーネルのオーバーヘッドと有益性との関連で検討が必要である。

4.3 割込み管理

chg_icr, ref_icr は、アプリケーション側からはサポートされると助かるが、プロセッサ / 割込みコントローラによってはサポートできない可能性がある。ITRON 仕様はハードウェアには依存しない方針なので、この機能を実現する場合に、ハードウェアに依存しないか、半導体メーカー側によるチェックが必要。

5 μ ITRON4.0 仕様との整合性

sta_tsk の起動コードの扱いや時間管理機能については μ ITRON4.0 仕様に整合させるのが適当と考えられる。

6 まとめ

原提案ならびに本ドキュメントで定義される仕様を、 μ ITRON4.0 仕様の 1 つのプロファイルとすることを、ITRON プロジェクトに対して、RTOS 自動車応用技術委員会から提案する。

以上