

## 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1回 組み込みシステムと並行性

皆様、はじめまして。大阪電気通信大学大学院の南角 茂樹（なんかくしげき）と言います。今回から6回ほどの予定でウィンドリバースクエアに投稿することになりました。よろしくお願いします。

最初に私とオペレーティングシステム（OS）やリアルタイム OS（RTOS）との関係を少しお話させていただきます。

私がインダストリアル系では業界トップクラスの総合電機メーカーに入社したのは1982年でした。大学でもマイクロプロセッサを使用した研究を行っていたため、会社でもマイクロプロセッサを使用した製品（今でいうところの組み込みシステムです、まだ組み込みシステムという言葉もなかったころから、ずっとそれにかかわってきたことは、私の誇りでもあります）の開発を行いたいと入社面接などでも訴えてきたかきもあり、配属は希望通りの工業用コントローラである NC 装置（Numerical Controller）という組み込みシステムを開発している部署でした。

ところで、組み込みシステムというと、今でこそ液晶テレビやカーナビゲーションシステムそして携帯電話、車などが代表としてあげられることが多いのですが、私がか社に入った80年代の初めのころは、インダストリアル機器が組み込み技術の最先端でした。

1982年当時でも NC 装置は CPU は i8086 を複数個使用したマルチ CPU システムでしたし、ほどなく i8087 という浮動小数点プロセッサの使用も開始しました。32ビット CPU であるモトローラ社の MC68020 を使用したのも世界的に見ても5番目以内でした。

ソフトウェアでいうと、すでに商用（New York 製です）のリアルタイム OS（おそらく世界最初の商用 RTOS です）を使用していました。ただし VxWorks とは異なり、この RTOS はオールアセンブラでしたし、VxWorks 同様 I/O もサポートしていましたが、そのためにはその RTOS が提供する色々な関数をいろいろ呼び出さなければならないという制約がありました。

その後1988年から CPU の変更や Unix との親和性、ネットワーク機能の必要性などから RTOS の変更の検討を開始して VxWorks と巡り合い、NC の関連製品で使用してその性能機能を確認後主力製品の NC 装置での全面的な使用に踏み切りました。ですから VxWorks とのかかわりもすでに20年を越えているわけです。

NC 装置などインダストリアル製品をはじめ、携帯電話や自動車、カーナビゲーションシステム、宇宙機器、非接触型 IC カード、自動販売機、家電製品、レーダー、エレベーターなど世の中は組み込みシステムで満ち溢れています。では組み込みシステムとはなんのでしょうか？その特徴の一つはソフトウェア、ハードウェア（デバイス）、メカ（機構）が一体となって、現実世界と相互作用を行っているシステムと言えるのではないのでしょうか。

以後ソフトウェアに話を絞ると、製品ソフトウェアは大きく、組み込みシステムに使用される組み込みソフトウェアと給料システムや履修システムなどの業務系ソフトウェア（エンタープライズ系などとも呼ばれることも

多いです)とに区別できると思います。(実はこれ以外にオフィス系など個人用ソフトウェア、OSとしてはTSS(Time Sharing System)もありますが、ここでは話の対象から外します)

組込みソフトウェアは次に何が起こるか分からない現実世界を対象にする場合が多いです。たとえば携帯電話であれば、いつメールが来るかわからない、その最中にいつ着信があるかわからない状況です。そして会話中にもCDMA方式のため送信電力制御を1ms程度の周期で(これもタイマー割り込みという一種の外部環境の変化ととらえることができます)実行しなければなりません。

また現実世界は同時に複数の変化が発生する場合もあれば、その重要性もさまざまであり、他の処理中であってもそれを一時停止しても、対応しなければならない重要な処理もあります。

イベントが発生してから演算処理を行い、結果を返すまでの時間制約があるものが組込みシステムには多く(それを守ることをリアルタイム性の実現と呼びます)、それを守ることが非常に重要です。以上をまとめると現実世界の変化の特徴は次のようになります。

- 現実世界の変化は非決定的に発生する
- 現実世界の変化への応答には時間制約がある
- 外部環境の変化は多様かつ複数の変化が同時に発生する可能性がある

これらの現実世界への応答を効果的に処理するためのソフトウェア構造としては、メインループの中で判定を行い、その結果に応じて処理を呼び出すという構造にすると、

- 処理周期が長くなる
- 処理中のより重要な処理に対する応答が難しくなる
- ソフトウェア的にも処理構造が複雑になり高品質なソフトウェアにするのが難しくなる

などの問題があります。

そこで図1に示すように外部環境の変化(イベント)に応じた個々の処理(プログラム)を作成しておき、イベントの発生に応じて処理を呼び出すような構造にしておくと、個々の処理が単純になりソフトウェアの品質も高いものにしやすくなります。

現実には、優先度の処理やプリエンプションなどの処理も必要になるため、アプリケーションプログラムはタスクという並行実行の単位で実行させ、それを制御するための図2に示すようにRTOSを導入するわけです。つまりRTOSの最大の目的は組込みシステムのリアルタイム性(システムによって決まっているデッドラインまでに応答を返すこと)の実現です。

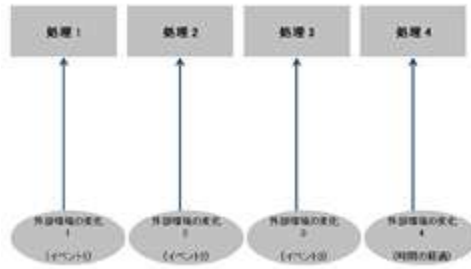


図 1.外部環境の変化に応じた処理

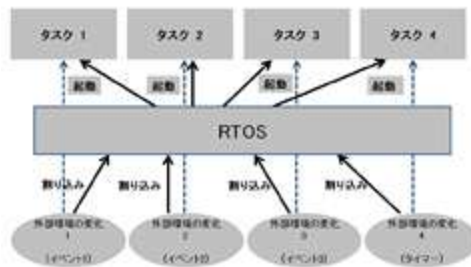


図 2.外部環境の変化と RTOS

一方業務系システムにおいて、その実行中に外部環境の変化に対応する必要性はほとんどありません。そして業務系システムにおいて最も求められるのは、処理量（スループット）が大きいことです。（ちなみに個人用 TSS で最も求められる特徴はなんでしょう？データやプログラムの保護機能でしょうか？）つまり OS といっても、組込みシステム用の RTOS に求められる機能と業務系システム用の OS に求められる機能は異なるということです。

これ以外も含めて RTOS に求められる機能を次にまとめます。

- リアルタイムシステムを構築するための OS
  - リアルタイム性を備えたシステムを構築しやすい
  - リアルタイム OS を使えばリアルタイムシステムが構築できるわけではない
  - きちんとしたリアルタイム設計は必要
- システムの時間的複雑度を下げる
  - 静的解析ツールが関数の複雑度を下げることによりソフトウェアの品質を上げる
  - のと同様に、時間的複雑度を下げることにより、リアルタイム性に起因する不具合の発生を防止してシステムの品質を上げる
- 組込みシステムにおけるプラットフォーム

第 1 回おわり

次回予告：デバイスドライバ等と密接に関連する、OS への通知手段である割り込みについて解説します。

## コラム：並行処理と並列処理

---

- 並行処理：英語では“Concurrent Processing”  
タスクや割り込みレベルで同時多重処理を実現するための技術。並行処理と言ってもソフトウェアレベルでは本当に複数の処理を実行しているわけではない。アプリケーション以外のソフトウェア（システムソフトウェア、OS）が処理を（高速に）切り替えることにより実現。
- 並列処理：英語では“Parallel Processing”  
CPUの命令レベルで同時多重処理を実現するための技術。実際に複数の処理を同時に実行している。

ただし両者を区別なく使用されることも多い。