

# 実時間処理システムの動作解析 / 評価ツール -- RPA(Real-time Performance Analyzer) --

1 K-9

大菅 与志一\*, 佐藤 真\*, 鈴木 繁\*, 古城 隆\*\*

\* 日本電気技術情報システム開発(株)

ソフトウェア(専) 第三開発部

\*\* 日本電気(株)

C&C 共通ソフトウェア開発(本) 基本システム部

## 1. はじめに

近年、組み込みシステムにおいて、実時間処理を行なうアプリケーションが数多く開発されている。

しかし、実時間処理の性能評価を効率的に行なう手法やツールはまだ少なく、タスクのプライオリティ設定ミスに起因するスケジューリングの乱れや、タイミングミスによる誤動作の原因等を究明できる汎用的手法も確立されていない。

今回、実時間処理システムの動作を視覚的に表し、容易に解析/評価できるツールを開発したので報告する。

## 2. 実時間処理の評価における問題点

実時間処理システムでは割り込み処理や各タスク処理の速度が時間的要求を満たすものでなくてはならない。即ち、システムの負荷が高い状態でも、決められた時間内に処理したり、外部からの割り込み在一定時間内に反応できることが保証されなければならない[1]。

しかし、組み込みシステムの開発過程で、タイミング的なミス等により、時間的要求を満たせなくなることは少なくない。このような場合、システムの動作を実時間的にトレースし、原因を分析するのが一般的である。しかし、既存のツールでは収集データに必要な情報が含まれて無かったり、データの出力形式が人間が素早く理解するのに不適当だったりして、誤動作の原因を解析する作業を困難にしている。

実時間システムの解析には、どのような方法で実時間的にデータを収集するか、また、どのようにシステムの動作を表し、何を集計すれば解析や評価が簡単に行なえるかが問題となる。

## 3. 実時間処理システムの評価方法

本システムは、実機を用いて、ソフトあるいはハード的にトレースデータを収集した後解析/評価する事後評価方式を用いる(図1参照)。トレースデータの収集方法は、それぞれ長所や短所があるため複数の方法をサポートする。このため、RPA<sup>2</sup>はトレースデータの収集方法に依存しない構成にした。

トレースデータの表示方法は、コンテキストスイッチ等によってCPUが不連続となった時間のデータを表形式で表示したのではその時間的間隔等が理解しづらいため、横軸を時間、縦軸をタスクとした図で表す[2]。図の表示にはX Window<sup>3</sup>を用い、GUIにはOSF/Motif<sup>4</sup>を利用している。また単に表示するだけでなく解析作業を支援する機能も追加した。

以降では、これに必要なデータ収集方法と表示方法、及び解析機能について述べる。

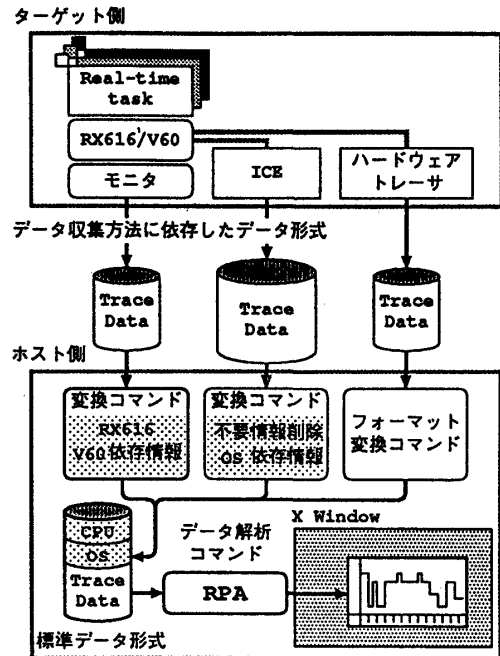


図1 システム構成

## 4. トレースデータの収集方法

### (1) ソフトウェアによる方法.

ターゲットシステム上のインタバル タイマ等を利用して、ソフト的に割り込みや例外の発生時刻を記録し、モニタを通してデータを収集する方法である。

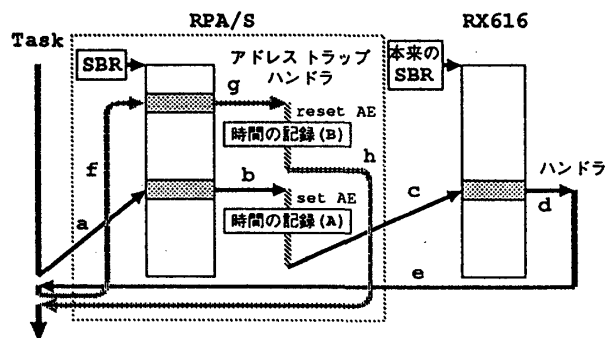


図2 V60でのトレースデータ収集方法

V60<sup>5</sup>でのデータ収集方法を図2に示す。図の処理はa, b, ...の順で進む。ハンドラに処理が移った時間は、システムベクタを2重化し検出する(図中のA)。処理が戻

Real-time Processing System Analysis/Evaluation tool.  
Yoshikazu OHSUGA\*, Makoto SATOH\*, Shigeru SUZUKI\*,  
Takashi KOJO\*\*

\* NEC Scientific Information System Development, Ltd.

\*\* NEC Corporation.

<sup>1</sup> ITRONの仕様に準拠したリアルタイム OS.

<sup>2</sup> システムの実時間解析ツール.

<sup>3</sup> X Windowは米国マサチューセッツ工科大学の登録商標です.

<sup>4</sup> OSF/MotifはOSFの商標です.

<sup>5</sup> NEC製32ビットMPU.

る時間の測定にはアドレストラップを利用する(図中のB)。このアドレストラップの設定(図中のset AE)は、開始時間測定時に行なう。この方式だとOSの構造を一切変える必要がない。

収集するデータは、時間、割り込み/例外の種類、走行中のタスク識別子、システムコールのアーギュメント等がある。1イベント当たりのデータはV60の場合、最大32byteである。

この方法では、イベント数に応じたオーバーヘッドが掛かる。V60でのオーバーヘッドはイベントの種類によって異なるが、MPUのクロックが16MHZのシステムで1イベント当たり60~100μsec程度であった。

同一イベントのオーバーヘッドは、ほとんど一定であるため解析時に考慮することが可能である。

(2) ハードウェアによる方法

ICE内のトレースデータやバストレーサのデータを利用する。この場合はトレーサに多量の高速メモリとホスト側に必要なデータを取り出すフィルタが必要となるが、オーバーヘッドは掛からない。

しかし、バストレーサを用いる場合にMPUに組み込まれたキャッシュメモリを利用するとデータが収集できないという問題点がある。

5. トレースデータの表示方法

実時間処理を解析するために有用な表示機能の例を図3に示す。

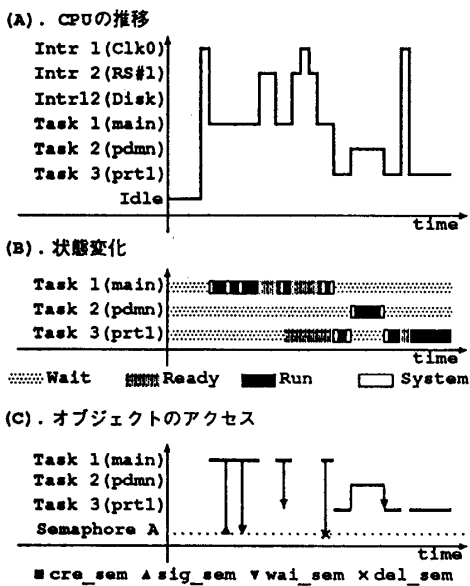


図3 実時間解析表示機能の表示例

(1) CPUの推移

CPUの移り変わりを直線によって表すものである。図3(A)の例では、縦軸が割り込みやタスク、アイドルであるが、もっと細かくタスク内の関数名でもよい。縦軸に何を表示するかは収集可能なデータと解析したい内容に依存する。

(2) 状態変化

状態の変化を色(模様)によって表すものである。図3(B)の例は、タスクの実行状態に着目した図である。

(3) オブジェクトのアクセス

オブジェクトのアクセスを記号で表すものである。図3(C)の例では、RX616のセマフォのアクセスを示している。オブジェクトとしてはセマフォの他にフラグやメール、タスク等が扱える。また、オブジェクトをメモリとし、そのメモリのアロケーションやリード/ライトアクセス、フリーなどを表すことも可能である。

6. 解析機能

解析機能は、トレースデータから必要な情報を効率的に得るための機能である。代表的な機能を2つ紹介する。

(1) 時間集計機能

RPAは、2つのカーソルを持ち、それらで指定した区間の時間的な割合を表示できる(図4参照)。

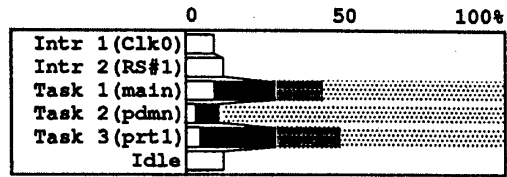


図4 閉区間の時間集計例

(2) 指定区間集計機能

指定区間とは2点の状態により定義することができる区間である。

例えば、割り込み処理、周期、割り込みが入らなかった区間は、それぞれ、その割り込み処理開始位置から終了位置まで、割り込み処理開始位置から次の開始位置まで、割り込み処理終了位置から開始位置までの各区間として定義できる。

この区間がデータ全体に幾つ存在し、時間的な分布がどうなっているかを示すヒストグラムを表示でき、最小/最大区間をデータ中から探すことができる。これにより最悪の場合のシステム動作解析が簡単に行なえる。

RPAには、この他に特定タスクや状態変化をサーチする機能等がある。

7. 考察

以上の方法により、今まで個々に評価や解析を必要としたシステムの実時間性能評価が総合的に可能になる。

例えば、割り込み処理の速度やCPUの使用率等が簡単に測定できる。タスクのCPU使用率とアイドルの比率から、そのシステム上での最大タスク数等を見積もることができる。

また、表示方法はターゲットシステムの動作を素早く理解することを支援する。例えば、マルチプロセッサシステムで起こったプライオリティインバージョン等のシステム動作は、ソースコードレビューでは発見しづらいが、本表示方法を用いれば視覚的に素早くその現象を発見できるようになる。

今後はプライオリティインバージョンが起こっている個所の発見を自動的に行なう等、汎用的な解析作業の自動化を進める予定である。

参考文献

[1]市瀬他, "リアルタイムOSにおける高速応答性能の実現と透過的OSモデルの実現", 情処第42回全国大会予稿集, (1991).  
 [2]H.Tokuda, "Arts real-time scheduler analyzer / debugger", IEEE Software, pp.29(May 1990).