実時間オペレーティングシステムR²用マルチタスクシミュレータの開発 R²シミュレータの機能と全体構成

5P-6

市岡 秀俊　大久保 英嗣　津田 孝夫　(京都大学工学部)
白濱 和人　佐々木 克也　日野 耕二　((株)ダイヘン)
宇都宮 敏行　(ダイヘンテック(株))

1. はじめに

R²シミュレータは、実時間オペレーティングシステムR²を用いたアプリケーションプログラムの開発支援システムD²の中核をなすソフトウェアであり、実時間システムのデバッグおよびテスト効率の向上を目的としている。

一般に、実時間システムは、以下のような要因によりデバッグおよびテストが困難であるとされている。

(1) 動作が非決定的である。すなわち、システム内のタスクのふるまいは内部状態にとどまらず、他のタスクや外部との相互作用によって決定される。

(2) ソースプログラムのみで、システム全体の正当性を検証することが困難である。すなわち、プログラムテキストの文脈のみでは他のタスクの命令の実行順序を知ることができない。

(3) テストケースに対応した実行環境の構築に時間がかかる。

本稿では、以上の問題点を解決するために開発を行ったR²シミュレータの機能と全体構成について述べる。

2. R²シミュレータの機能

R²シミュレータは、R²上に構築される複数のアプリケーションタスクの実行を開発システム上でシミュレートし、タスクに関連する各種情報をユーザにフィードバックする。R²シミュレータの主な機能として以下のものがある。

(1) マルチタスクデバッグ機能

オペレーティングシステムの管理下にあるデバッガでは、デバッグ中に非同期的な外部事象によってユーザの意図しない文脈でタスクスイッチが発生し、制御がデバッグ対象のタスクからシステム核に移行する。従って、タスクプログラムの文脈のみでシステム内のすべてのタスクの実行系列を把握することが困難である。R²シミュレータでは、論理時刻を使用して複数タスクの並行実行をシミュレートしているので、見かけ上、制御は遷移すべきタスクに直接移行するため、タスクの実行系列を把握することが容易である。

(2) シンボリックデバッグ機能

R²のアプリケーションプログラムは、C言語を使用して記述される。このために、R²シミュレータではC言語のソースコードレベルでの強力なシンボリックデバッグ機能を提供している。本シンボリックデバッグ機能では、マルチタスクプログラムのデバッグに対応するため、実行中のタスク以外のタスクのシンボルに対するアクセスを可能にしている。さらに、シンボルに対してスコープルールを適用しており、自動変数や静的変数を含めてタスク内で定義されているすべてのシンボルに対してアクセスすることができる。

(3) システムモニタリング機能

ユーザは、端末からコマンドを投入することによって、各タスクの状態や発生した割込みの状態を知ることができる。さらに、それらの状態を動的に変更することも可能である。また、指定したユーザ時刻(但し、R²シミュレータにおける論理時刻)に任意の割込みを発生させることも可能となっている。従来は、タスクの状態変更や割込みのトレースのために、ソースプログラムのエディット、コンパイル、リンクといった過程を経る必要があったが、本機能を使用することによってこれらのシステム再構築に要する時間が大幅に短縮されることになる。

(4) 入出力シミュレート機能

入出力のシミュレートを行うことにより、実時間システムでは困難とされている入出力処理のデバッグを可能にしている。入出力シミュレート機能には、入出力コントローラと入出力命令の2つのシミュレート機能がある。前者は、割込みコントローラを含めた入出力コントローラの動作をシミュレートするものである。入出力コントローラの動作は、ユーザが入出力コンフィギュレーションファイルと呼ばれる初期化ファイルに記述する。後者は、入出力ポートに入出力コントローラが接続されていない場合の動作をシミュレートする機能である。これらの機能を使用することによって、実際に機器を接続することなく割込みや入出力といった外部事象に関する処理のためのテストを行うことができる。

R²シミュレータでは、以上の4つの機能の他に、関数単位の実行時間等のプロファイル情報の取得の

The Development of Multi-Task Simulator for Real-Time Operating System R²
Features and Software Configuration of R² Simulator
HIDETOSHI ICHIOKA, EIJI OKUBO and TAKAO TSUDA (Kyoto University)
KAZUTO SHIRAHAMA, KATSUYA SASAKI and KOJI HINO (Daihen Corporation)
TOSHIYUKI UTSUNOMIYA (Daihen Technology Institute)

ための機能、コマンドおよび入出力動作のヒストリ機能、一連のデバッグ手順を記憶させるコマンドプログラムの機能等も提供している。

3. R2シミュレータの全体構成

R2シミュレータの全体構成を図1に示す。このうち、R2シミュレータの実行フェーズに関与するモニタ、インタプリタ、R2核、I/Oシミュレータについて説明する。

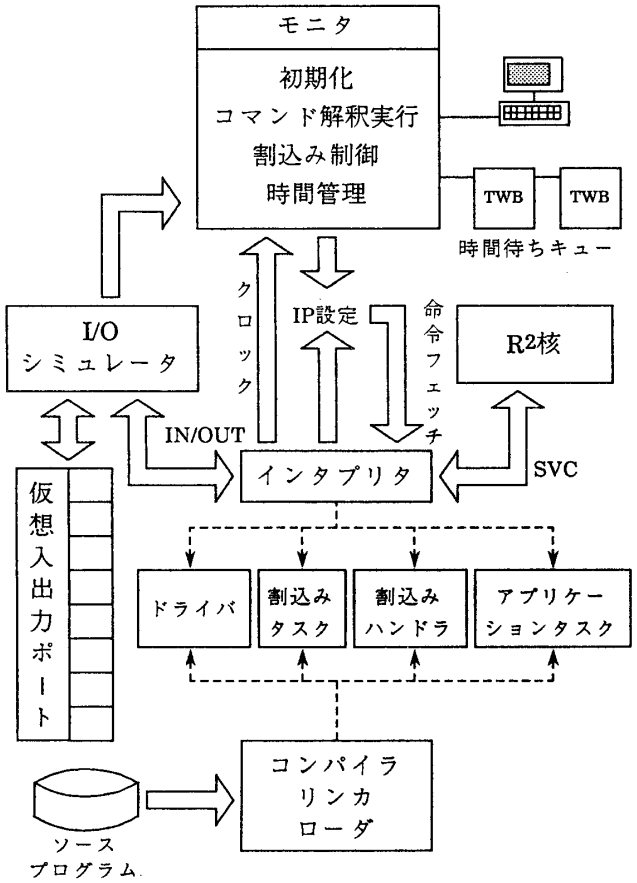

図1 R2シミュレータの全体構成

(1) モニタ

モニタは、主に割込みの制御および時間の管理を行う。すなわち、R2の割込み処理をシミュレートし、タスクスイッチを管理する。インタプリタが1命令実行する毎に設定する処理時間に関する情報を得て、これをもとにシステム時刻とユーザ時刻とを更新する。これらは主にR2のタイムウェイト処理やデバッグ機能としてのプロファイル情報を得るために用いられる。割込みの検出は、モニタの時間管理部が行う。またデバッグ用のコマンドの解釈実行もモニタによって行われる。

(2) インタプリタ

R2シミュレータでは、仮想的な計算機(R2仮想計算機)を想定している。インタプリタは、R2仮想計算機上で中間コードを解釈実行する。中間コードは、シミュレータのための専用のコンパイラによって生成されるオブジェクトコードである。インタプリタは、中間コードのうち、SVC命令と入出力命令以外の命令の処理ルーチンから構成される。R2のSVC命令はR2核を、入出力命令はI/Oシミュレータを呼び出すことで実行される。割込み関連のインタフェース用の関数はモニタが処理する。その他、C言語のライブラリ関数(例えば、printfやsin等)は、UNIX(VMS)で提供されている実行時ライブラリを利用している。インタプリタは、各命令の処理時間をテーブルの形で保持しており、1つの命令を実行する毎に、R2仮想計算機のインストラクションポインタを更新するとともに、実世界での対応する処理に要する時間を反映できるように時間情報を設定する。しかし、一般にSVC命令の実行などにおいては、条件により実行時間に差が生じる場合がある。その対応はデバッグ時にユーザに委ねられ、そのためのコマンドを提供している。すなわち、絶対的な時間をシミュレートすることは不可能であり、相対的な進行順序が保証されれば十分であると考えている。

(3) R2核

R2核はR2のSVC処理ルーチンをはじめとして、R2で提供されているライブラリ群よりなり、インタプリタからSVCの処理を委ねられる。

(4) I/Oシミュレータ

I/Oシミュレータは、中間コードのIN/OUT命令に対応する処理を行う。本シミュレータでは、多種多様な入出力装置に対応するために、各入出力動作に関して、仮想的な入出力ポートのレベルで処理を行っている。すなわち、IN/OUT命令に関しては、それぞれ図1のように仮想入出力ポートからの読み出しおよび仮想入出力ポートへの書き込みを行う(但し、当該入出力ポートの条件を満足している場合にのみ、対応する機能を実行する)。また、入出力の完了等のハードウェア割込みはI/Oシミュレータによって起動される。入出力結果は、入出力動作のヒストリとしてデータベースに蓄積される。これは、多くの入出力処理においては、1つの入出力ポートに関して複数のデータのやり取りがあった後に1つの処理が完了するためである。

4. おわりに

以上、R2シミュレータの機能と全体構成について述べた。現在、UNIXおよびVMS上に第1版が完成しており、その評価を行っている段階である。今後は、第1版のチューンアップ、さらに分散環境におけるテストに対応したR2シミュレータの分散バージョンの開発に取り組む予定である。

参考文献

[1] F. Baiardi, N. De Francesco, E. Matteoli, S. Stefanini, and G. Vaglini: "Development of a Debugger for a Concurrent Language", SIGPLAN Notice, Vol. 18, No. 8, pp. 98-106 (1983).