

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

5P-7

R²シミュレータの言語処理系

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))
 市岡 秀俊 大久保 英嗣 津田 孝夫 (京都大学工学部)

1. はじめに

一般に、産業用ロボットやNC工作機械をはじめとする実時間制御システムにおいては、システムの開発環境と実行環境が異なる場合が多い。しかも、システム全体のテストを行うためにはどうしてもテスト項目に対応した実機テストに頼らざるを得ないのが現状である。また、実行環境におけるデバッグ手段として、ICEをはじめとする各種ツールが使用されているが、このようなICEを主体とした従来のデバッグにおいては、割込みの制御や入出力ドライバ等の非同期に実行されるソフトウェアのデバッグに非常に時間がかかっていた。さらに、レジスタや各種のフラグ等を意識した機械語レベルのデバッグであるため、プロセッサや機械が変更される度にそれらの詳細な知識の修得が必要となる。

我々は、実時間制御システムに組み込み可能な実時間オペレーティングシステムR²の開発を行ってきた[1]。R²プロジェクトでは、このような実行環境で行われているテストを、開発環境においてもサポートするためにR²用マルチタスクシミュレータ(以下R²シミュレータと略す)の開発を行っている。本稿では、シミュレーションシステム構築のためのツールであるR²シミュレータの言語処理系に関して、その特徴と構成について述べる。

2. R²シミュレータにおける言語処理系の特徴

R²シミュレータは、ユーザによってC言語で記述されたアプリケーションプログラムの実行をシミュレートするインタプリタとして実現されている。R²シミュレータにおける言語処理系の特徴として、以下の4つが挙げられる。

(1) C言語によるマルチタスク処理の記述

R²のアプリケーションプログラムは、入出力ドライバを含めて、すべてC言語で記述することが可能となっている。R²シミュレータで提供されるCコンパイラ(R²CC)では、R²の各SVC命令を1つの中間言語に変換する。しかも、インタプリタ(R²SIM)はそれらのSVC命令をR²の仕様に沿って忠実に実行する。従って、シミュレーションを行う場合、ユーザはシミュレートすべきアプリケーションプログラムを変更することなく、これらのSVC命令を使用して、タスクの生成や消滅、およびタスク間通信等の

マルチタスク処理を記述することが可能となっている。

(2) タスクの中間言語レベルでのシミュレーション

R²シミュレータは、中間言語のレベルでアプリケーションプログラムの実行をシミュレートする。R²シミュレータの中間語の形式は、命令を表すcodeの部分と命令の処理対象となるop1およびop2とからなる3つ組(code, op1, op2)の構成となっている。現在、R²シミュレータでは約200個の中間言語の命令が提供されている。アプリケーションプログラムをこのような中間言語のレベルで解釈実行することにより、機械語レベルでのシミュレーションに比べて、シミュレーションの実行時間を短縮することが可能となる。さらに、機械語を意識することなくC言語のソースレベルでのデバッグが可能となりデバッグ効率が向上する。

(3) アプリケーションタスクおよび入出力ドライバの登録

R²シミュレータでは、ユーザにより作成された任意のアプリケーションプログラムをR²のタスクおよび入出力ドライバとして登録する機能をサポートしている。本機能を使用することによって、テスト項目に対応したマルチタスク環境を、開発環境において容易に構築することができる。従って、実行環境におけるテストに比べ効率の良いテストおよびデバッグが可能となる。

(4) 入出力ドライバとR²核の動的リンク

通常、機器組み込み型の実時間オペレーティングシステムにおいて入出力ドライバのテストを行う場合、それらをオペレーティングシステム核とオブジェクトファイルのレベルでリンクしなければならない。これに対して、R²シミュレータでは、この入出力ドライバとR²核とのリンクを、ローダに組み込まれている入出力コンフィギュレータ(IOCONF)を使用して、シミュレータ起動時に動的に行っている。入出力ドライバとリンクの必要なR²核は、入出力管理部等の一部であるため、この方式が可能となっている。本方式により、オブジェクトファイルのレベルでのリンクに比較すると、入出力ドライバのテスト効率が著しく向上すると考えられる。

3. R2シミュレータにおける言語処理系の構成

R2シミュレータの言語処理系は図1に示すように各種のソフトウェアから構成されている。以下、これらの構成要素について説明する。

(1) Cコンパイラ(R2CC)

R2CCは、C言語のソースプログラムを中間言語に変換する。R2シミュレータは、中間言語としてR2のSVC命令をサポートしている。このためR2CCは、字句解析部の内部データとしてR2がサポートしているSVC命令の表を持ち、これを用いてR2のSVC命令を認識している(実際には、R2のSVC命令は予約語として予め予約語テーブルに登録されている)。字句解析部により認識されたR2のSVC命令は、構文解析部のコード出力部により中間言語として出力される。R2CCは、以上の中間言語とともに、デバッグに必要となる各種情報をオブジェクトファイルとして出力する。オブジェクトファイルには、コンパイルの対象となっているソースファイル名、当該ファイル内で定義されている関数や外部変数に関する情報等が格納される。

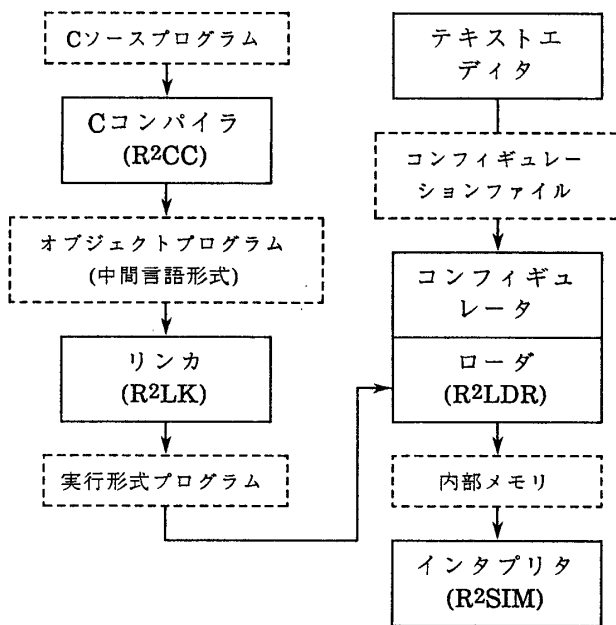


図1 R2シミュレータの言語処理系

(2) リンカ(R2LK)

R2CCにより出力されたオブジェクトファイルは、リンカ(R2LK)により単一のリロケートブルな実行ファイルとしてまとめられる。

(3) ローダ(R2LDR)

R2LKにより出力された実行ファイルは、シミュレータ起動時にローダ(R2LDR)によりシミュレータの内部メモリ上に読み込まれ、R2のアプリケー

ションタスクおよび入出力ドライバとして登録される。R2LDRには、タスクコンフィギュレータ(TCONF)と入出力コンフィギュレータ(IOCONF)が組み込まれている。

(4) タスクコンフィギュレータ(TCONF)

タスクコンフィギュレータ(TCONF)は、ユーザにより作成されたアプリケーションプログラムをR2のタスクとして登録するためのものであり、タスクコンフィギュレーションファイル(TCF: Task Configuration File)の情報をもとに当該タスクの登録を行う。TCONFの主な設定項目には、ルートタスクのファイル名およびタスク番号、ルートタスクを配置するCPUの番号、ルートタスクの優先度、スタックサイズ、タスクのメイン関数名等がある。

(5) 入出力コンフィギュレータ(IOCONF)

入出力コンフィギュレータ(IOCONF)は、ユーザにより作成された入出力ドライバをR2の入出力ドライバとして登録するためのものであり、入出力コンフィギュレーションファイル(IOCF: I/O Configuration File)の情報をもとに当該入出力ドライバの登録を行う。IOCONFの主な設定項目には、ドライバのファイル名、デバイス数およびデバイス名、デバイス初期化ルーチン名、SVCルーチン名、タイムアウト処理ルーチン名、入出力コマンドの実行時間などがある。

(6) インタプリタ(R2SIM)

インタプリタは、R2のSVC命令と入出力命令以外の中間言語を解釈実行する。R2のSVC命令は、R2核の内部関数を用いて関数呼び出しとして実現されている。IOシミュレータは、仮想的な入出力ポートを有しており、中間言語のIN/OUT命令は、この仮想入出力ポートへの読み出しおよび書き込み処理として実現されている。

4. 言語処理系の開発環境

R2シミュレータの言語処理系は、VAX-11/750 (UNIX 4.2 bsd)および μ VAX (VMS)上で開発されている。また、すべての言語処理系はC言語を用いて記述されているため、このような異なるオペレーティングシステム間の移植作業も容易に実現されている。

5. おわりに

以上、R2シミュレータの言語処理系に関し、その特徴と構成について述べた。今後は、本シミュレータを各種実時間システムの開発に適用して総合的な評価を行っていく予定である。

(参考文献)

[1] 大久保, 他: 実時間オペレーティングシステムR2の設計目標と全体構成, 情報処理学会第33回全国大会講演論文集, 3V-6 (1986).