

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

3V-6

大久保 英嗣 津田 孝夫 楠田 修三 小林 正典(京都大学工学部)

杉村 邦彦 白濱 和人 友田 和伸((株)ダイヘン)

1. はじめに

R²は、ロボットマニピュレータやNC工作機械等のロボットシステムにおける各種制御装置に組み込み可能な実時間オペレーティングシステムである。R²は、従来の制御用ソフトウェアにおける設計、開発及び保守時の問題点を解決するために開発された。即ち、R²は制御用ソフトウェアの応答性と移植性の向上を目標としている。さらに、アプリケーションプログラムの開発を容易にするためのユーザ親和性を持ったインタフェースを実現するよう設計されている。

R²は、現在、マイクロプロセッサ8086上に構築されている。我々は、これをR²-86と呼んでいる。R²-86の実現は、主にC言語を使用して行われている。将来的には、マイクロプロセッサ68000をもサポートする予定である。本稿では、R²-86第1版の設計目標及び全体の構成について述べる。

2. 設計目標

一般に、実時間制御を初めとして各種制御用ソフトウェアには、以下のような要求が存在する。

- (1)特定の事象、即ち割込みに高速に応答可能であること。
- (2)タスク間通信あるいはタスクスイッチ等のシステムオーバーヘッドが小さいこと。
- (3)種々の入出力機能が標準の機能としてサポートされていること。あるいは、ユーザが容易にそれらの機能をプログラム可能なこと。
- (4)生産ラインの再編成に対応して、各装置の制御用ソフトウェアも再構成可能なこと。

これらの要求に答えるために、R²では、実時間システムに要求される高速応答性の達成のみならず、容易に移植可能な(そして保守可能な)システムの実現を主目標としている。そのため、R²では、高速応答性に関して割込み処理とタスクディスパッチング処理の高速化を実現している。さらに、実時間OSでは十分にサポートされていないファイル管理の機能を標準でサポートしており、ディレクトリを主記憶に常駐させることによって、2次記憶の高速アクセスを実現している。一方、移植性に関しては、アプリケーションプログラムの可読性を向上させ、さらにソフトウェアの再利用を可能とするために、“ロ

ボットプリミティブ”と呼ばれるC言語の関数を基本としたシステム構成法を採用している。さらに、機械依存部を除いてR²自体もC言語で記述されている。また、機械依存部に関しては、割込みハンドラや入出力ドライバを記述するためのC言語インタフェースを持つ各種ライブラリを提供している。

3. R²-86の全体構成

R²-86は、マイクロプロセッサ8086/8088/80186上で動作可能な実時間OSである。但し、ハードウェアとしてインターバルタイマ及び割込みコントローラの装備を前提としている。R²-86は、RAMベース及びROMベースの両方の型のシステムとして生成することが可能である。R²-86は、主にロボット制御を対象に考えており、R²-86標準ではロボットのティーチングデータの管理やロボットプリミティブ等のロボットに関する処理の専用機能を有している。しかし、システムの基本部分であるR²-86核は実時間制御が行われる8086ベースの任意のシステムで使用可能である。

R²-86システムは、以下のソフトウェアから構成される(図1参照)。

- (1)R²-86核
- (2)入出力ドライバ
- (3)ファイル管理
- (4)ロボットプリミティブ
- (5)ファシリティ(アプリケーションプログラム)
- (6)ユーザインタフェース
- (7)デバッグ及びシミュレータ

特に、本構成法と従来の構成法の相違点は、ロボットプリミティブを設定した点にある。以下、各構成要素を簡単に説明する。

(1)R²-86核

R²-86のOS核は、割込み、タイマ制御、入出力等の外部事象のための制御プログラムと、タスク管理及びメモリ管理等の内部事象の制御プログラムに大別される。さらに、実時間システムでは高速応答が要求されるので、タスクのスケジューリングの他に、割込み自体のスケジューリング機能が組み込まれている。

The Design Objectives and Architecture of Real-Time Operating System R²

ELJI OKUBO, TAKAO TSUDA, SYUZO KUSUDA and MASANORI KOBAYASHI (Kyoto University)

KUNIHICO SUGIMURA, KAZUTO SHIRAHAMA and YASUNOBU TOMODA (Daihen Corporation)

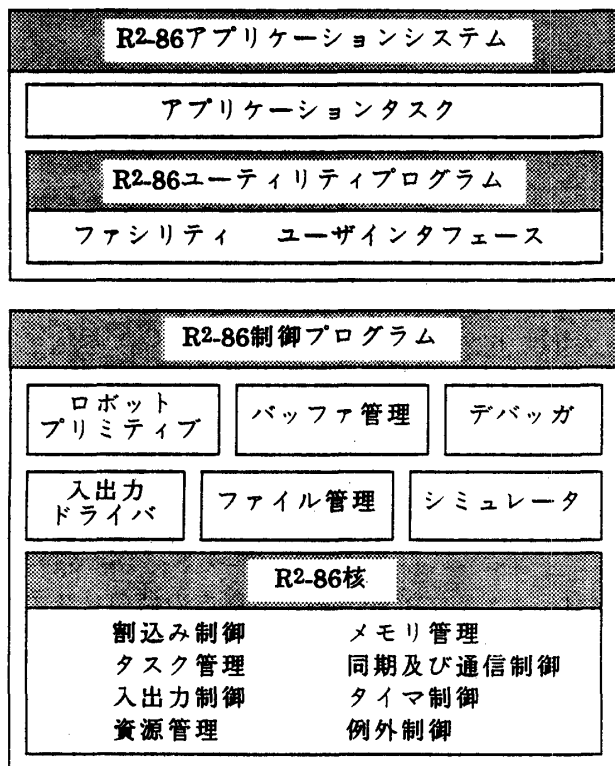


図1 R2-86のソフトウェア構成

(2) 入出力ドライバ

ハードウェアからの独立性を向上させるため、即ちハードウェア変更に伴うアプリケーションプログラムの変更を緩和するため、各種入出力をすべて統一して扱えるインタフェースを提供している。具体的には、バイトやビット単位の入出力を行うディスクリット入出力、A/D及びD/A、CRT、磁気バブル、さらに通信のための入出力を一括してSVC処理として扱っている。但し、ビット単位の入出力に関しては、仮想的なSVC処理を設定している。

(3) ファイル管理

R²のファイル管理では、プログラム開発用の小型OSであるMS-DOSのファイル構造(階層的ディレクトリ構造)を管理する機能をサポートしている。さらに、バッファ管理機能及びティーチングデータの管理機能も標準でサポートしている。ティーチングデータは、ロボットの動作を記述するものであり、一般にフロッピーディスクや磁気バブル等の外部記憶に格納されている。R²-86では、このティーチングデータに対するランダムアクセスとインデックスによるアクセスをサポートしている。

(4) ロボットプリミティブ

R²では、これまでに開発してきたロボット制御用プログラムから共通的かつ普遍的な機能を抽出し、ロボットプリミティブとしてまとめている。これには、ロボットの動作プリミティブ、座標変換

用プリミティブ、ティーチング用プリミティブ等がある。

(5) ファシリティ(アプリケーションプログラム)

R²では、ロボット用アプリケーションプログラム(データ編集、異常処理、ティーチング、ブロック運転、自動運転等)をファシリティと呼び、これらのプログラムをロボットプリミティブの系列を使用して構成している。これにより、ファシリティレベルの移植性が大幅に向上すると考えられる。

(6) ユーザインタフェース

ロボット言語用コンパイラを初めとしたオフラインプログラミング機能、ティーチングボックスによる教示、実行状況の表示及び稼働管理情報の編集・出力といった生産管理等のプログラムがある。

(7) デバッガ及びシミュレータ

R²-86では、システムのモニタリング及びデバッグ機能を内蔵している。さらに、ターゲットプロセッサの動作をシミュレートするためのソフトウェアシミュレータをR²ファミリとしてサポートすることを予定している。これらの機能により、アプリケーションプログラムを作成するための専用のプログラム開発装置が不要になると考えられる。

4. R2の開発環境

R²は、VAX-11/750 (UNIX 4.2bsd)上で開発されている。VAX上では、各モジュールのコンパイル、モジュール間インタフェースの整合性チェック、各種ツールを利用したテストを行っている。特に、複数モジュールの結合テストに関しては、R²専用の環境を構築するための小さなシミュレータを使用している。さらに、ロボットシステムの実行システムとしては、ムーブマスターIIをPC9801に接続したシステムを使用している。R²自体のシステム生成は、VAXとPC9801間でソースレベルのファイル転送を行い、PC9801上のMS-DOSを使用して行っている。

5. おわりに

以上、本稿では応答性と移植性を目標とした実時間制御用のOSであるR²-86について述べた。今回実現されたR²-86の大部分はC言語により記述されており、当初の目的である移植性の高いシステムの実現は十分に達成されたと考えている。今後は、R²-86を使用した各種アプリケーションプログラムを開発し、評価を行っていかねばならないと考えている。このため、現在、2台のPC9801とロボットコントローラ及びロボット本体各1台のプロトタイプシステムを構築中である。

応答性に関しては、他の実時間OSとの比較も含めて現在性能評価を行っている段階である。