3V-9

実時間オペレーティングシステムR²の入出力制御方式

白濱 和人　杉村 邦彦　友田 和伸 ((株)ダイヘン)
大久保 英嗣　津田 孝夫　楠田 修三　小林 正典(京都大学工学部)

1. はじめに

産業用ロボットシステムをはじめとして、実時間制御システムにおける入出力処理は多岐に渡っており、これら入出力の処理速度およびスケジューリングはシステムの性能を決定する上で重要な要因の1つとなっている。今回、我々は実時間オペレーティングシステムR²を開発するに当たって、このような多種多様の入出力処理の最適制御と入出力装置の仮想化を実現することにより、実時間制御システムの開発と保守における当事者の負担の軽減を図った。

本稿では、R²における入出力制御の特徴ならびに方式について述べる。

2. R²における入出力制御の特徴

R²の入出力制御は実時間制御システムにおける入出力管理の重要性および特徴を考慮して設計されている。即ち、

(1) 複数の入出力処理の同時実行を可能とする。

(2) 入出力ドライバの作成を容易にする。

(3) 入出力装置を仮想化させる。

の3つである。

タスクから入出力要求があると、制御は入出力ドライバに移り入出力装置が起動される。ドライバ内では当該入出力の完了を待つことなく、直ちにタスクに制御が戻る。従って、タスクは入出力動作が完了するまで他の処理を並行して実行することが可能である。またこの時、当該タスクより優先度の高いタスクが存在すれば、R²はそのタスクを優先してスケジュールする。スケジュールされたタスクは、また別の入出力を要求することができる。既に動作中である入出力装置に対して新たな入出力要求が発行された場合、当該入出力装置に対する入出力要求のキューが作成され、入出力装置が解放されるまでその要求は待たされる。このようにR²では、複数のタスクと複数の入出力を並行に実行することによって、システムのスループットを向上させている。

入出力ドライバの作成を容易にするために、R²ではドライバ自体をC言語で記述することを可能としている。さらに、記述されたドライバとR²核とのインターフェースのために各種ライブラリ関数を用意している。

また、R²ではアプリケーションプログラムから入出力要求を行う場合、そのためのパラメータとして、入出力装置の論理番号、入出力データを格納するバッファへのポインタ、動作の種類を指定する機能番号等、抽象化された値のみを設定すれば良く入出力装置の仮想化を実現している。従って、ハードウェアを変更する場合でも、直接アプリケーションプログラムを修正する必要はない。これにより入出力ドライバとアプリケーションプログラムの設計を独立に並行して進めることができ、入出力ドライバのみならずアプリケーションプログラムの移植性をも高めている。

3. R²における入出力制御方式

R²では、3種類の入出力機能をサポートしている(表1参照)。以下、各機能について説明する。

表1 入出力システムコール一覧

	サービス名	サービス内容
SVC	sio()	標準入出力装置に対する入出力要求のスケジューリング
	dio()	ディスクリート入出力装置に対する入出力要求のスケジューリング
実行時ライブラリ	bi()	入出力ポートの直接アクセス(バイト入力)
	bo()	入出力ポートの直接アクセス(バイト出力)
	wi()	入出力ポートの直接アクセス(ワード入力)
	wo()	入出力ポートの直接アクセス(ワード出力)

3.1 標準入出力(sio())

CRT、キーボード等R²で標準的にサポートしている入出力装置に対するアクセスを行ったり、ユーザシステムに固有な一連の入出力操作を一まとめにして実行したいとき等、入出力動作の実行をR²の管理にまかせる方式である。この場合R²のユーザは、入出力ドライバとして、以下の各ルーチンを登録する必要がある(但し、ユーザシステムの応用によっ

The Architecture of I/O Control in Real-Time Operating System R²

KAZUTO SHIRAHAMA, KUNIHIKO SUGIMURA and YASUNOBU TOMODA (Daihen Corporation)
EIJI OKUBO, TAKAO TSUDA, SYUZO KUSUDA and MASANORI KOBAYASHI (Kyoto University)

ては不要のものもある)。なお、これらのルーチンのエントリアドレスはシステム構築時に、ユーザによって定義される。

(a)入出力装置の初期化ルーチン

システムの立ち上げ時にR²が自動的に呼び出して実行するルーチンであり、該当する入出力装置の初期化を行うものである。

(b)入出力要求のサービスルーチン

タスクが入出力装置に対して入出力要求を発行したとき、R²が呼び出すルーチンであり、当該入出力装置に対する実際のアクセスが記述される。

(c)入出力装置からの割込み信号処理ルーチン

割込み信号が発生する入出力装置に関して、割込み発生時の処理を記述するルーチンである。このルーチンでは、割込みが発生した時に、さらに入出力動作を継続するか、あるいは入出力動作の完了を入出力要求を行ったタスクに知らせるかを記述する。但し、これは各ユーザシステムの応用によって決める。

(d)デバイスタイムアウト発生時の処理ルーチン

入出力動作完了の時間監視を行う場合に記述する。このルーチンは、指定された一定の時間を経過しても入出力動作が完了しない場合にR²が呼び出すルーチンである。

3.2 ディスクリート入出力(dio())

ディスクリート入出力を行う場合、一連の入出力操作が複数の入出力ポートに及ぶことがよくある。これに対して、R²では最大4つの8ビット物理入出力ポートを、連続した1ワードの論理入出力ポートに変換することによって、入出力の記述を一まとめにするための機能を提供している(図1参照)。さらに、この入出力に関してはsio()と同様に、複数のタスク間で排他的にアクセスする機能をも実現している。

R²のディスクリート入出力管理機能として、以下に示す3つがある。

(a)ディスクリート入力

ディスクリート入力は、論理入力ポートからの入力値に、必要なビットマスクをかけユーザの指定したバッファに格納する。

(b)ディスクリート出力

ディスクリート出力は、ユーザの指定するバッファの内容に必要なビットマスクをかけ論理出力ポートに出力する。この時、マスクされないビットの出力値は、前の値が保持される。

(c)ディスクリート出力値の確認

ディスクリート出力値の確認は、指定された論理出力ポートへの現在出力値に、必要なビットマスクをかけユーザの指定したバッファに読み出す。

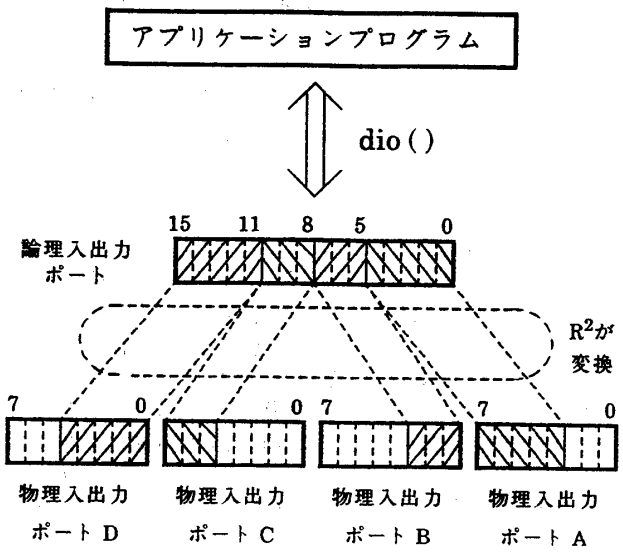

図1 論理入出力ポートの構成

3.3 入出力ポートの直接アクセス

実時間制御システムにおいては入出力の速度が重要で、入出力毎のSVCのオーバヘッドが問題となる場合もある。また、入出力ドライバをC言語で記述する場合、入出力ポートへの入出力操作もC言語で記述できなければならない(標準C言語では、この機能はサポートされていない)。これらに対して、R²では簡単なディスクリート入出力は、アプリケーションプログラムおよび入出力ドライバの中で、物理入出力ポート番号を直接指定して入出力を行うことも可能としている。即ち、以下のように記述することができる。

```
input_data = bi(port_no);     /* バイト入力 */
input_data = wi(port_no);     /* ワード入力 */
bo(port_no, output_data);     /* バイト出力 */
wo(port_no, output_data);     /* ワード出力 */
```

ここで、port_no、input_data および output_data は、C言語の unsigned 型の変数または定数である。port_no はディスクリート入出力ポートの物理番号、input_data は入力データ、output_data は出力データを各々示している。但し、バイト型入出力の場合は、下位8ビットのみ有効となる。

4. おわりに

以上、R²の入出力制御に関し、その特徴と方式について述べてきた。R²では3種類の入出力管理をサポートしており、多種類の入出力システムの要求に柔軟に対応することが可能となっている。今後、各種実時間システムに適用して総合的な評価を行っていく予定である。