

生産管理のシステム化

- リアルタイム・モニタの開発 -

機械金属部 中嶋 勝也 大力 賢次

近藤美智雄*

1. 緒言

多くの製造業においてその生産形態は、多品種少量になっている。このような中で、自動化・省力化が従来から進められ、生産コストの低減に役立っている。多品種少量生産を効率的に行うためには、生産管理が重要な役目を果たすが、そのためには、現在の生産ラインの状況を、正確に、早くとらえる必要がある。しかしこの部分での自動化は進んでおらず、人手によって行われているのが実状である。

そこで本年度より、生産管理のための情報収集を自動化するために、図1に示すようなシステムを開発することを目的に、研究を行っている。

本年度は、生産現場の状況を監視するシングルボード・コンピュータに必要となる、リアルタイム・モニタを開発したので報告する。

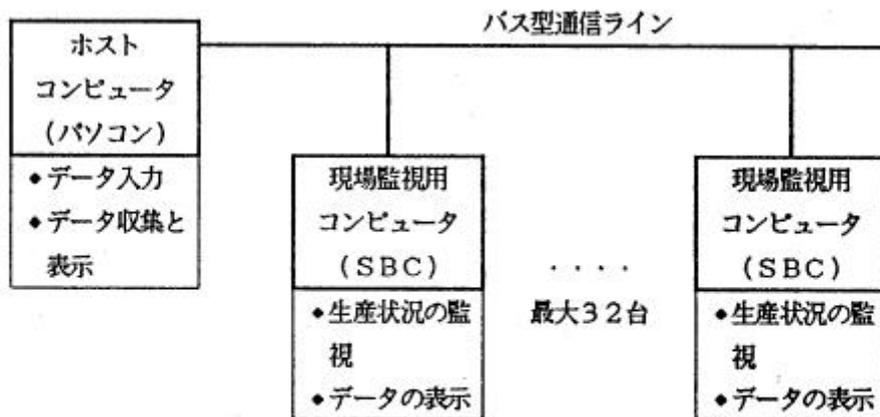


図1：生産管理システムの構成

2. IRTMの概要

開発したリアルタイム・モニタを、IRTMとよぶ。このモニタは、小規模のマイコン・システムで動作させることを前提に設計した、小型のリアルタイム・マルチタスク・モニタである。小規模のマイコン・システムでリアルタイム・モニタが必要になる場合は、互いに独立したタスクを複数動作させるというマルチタスク性が重要となるよりは、むしろリアルタイム性を必要とする一つの仕事を、複数のタスクに分けて設計するための開発環境として、リアルタイム・モニタを使用するケースが多いと考えられる。

* (有)コーヨーエンジニアリング

このような状況で使用されることを前提とし、IRTMの基本仕様を以下のように設定した。

最大 255 個のタスクを登録・管理することができる。

優先順位によるスケジューリングを行う。

16 種類のシステム・コールと 4 種類のマクロが用意されている。

- | | | | |
|---------------|------|---------------|------|
| a. タスク管理 | 9 種類 | b. イベント・フラグ管理 | 3 種類 |
| c. メールボックス管理 | 2 種類 | d. 時間管理 | 2 種類 |
| e. 割り込み管理用マクロ | 4 種類 | | |

リモート・モニタをタスクとして搭載することができ、各タスクのデバッグが行える。

また I RTM の開発にあたっては、TRON 協議会による I TRON 仕様を参考にし、システム・コールの名称や機能を、I TRON 仕様にできるだけ近づけた。しかし I TRON では、タスク管理、共有資源管理、メモリ管理等の面において、必ずしも小規模なマイコン・システムに必要ではないと考えられる機能も多く、これらの機能の多くを削除した。

これにより、モニタ本体（スケジューラおよび各システム・コール処理ルーチン）のサイズも、約 1K バイトと小型なものとなった。

さらに IRTM の開発にあたっては、従来から開発してきたリモート・モニタを、マルチタスク・プログラミングのためのデバッグとして搭載できるよう、合わせて開発を行った。

3. IRTM のタスク管理

I RTM では、最大 255 個のタスクを登録・管理することができる。これらのタスクは、さらに次に示す 6 つの状態のいずれかに属して管理され、図 2 のように遷移する。

未登録状態 (NON — EXISTENCE)

休止状態 (DORMANT)

実行可能状態 (READY)

実行状態 (RUN)

待ち状態 (WA I T)

強制待ち状態 (SUSPENDED)

IRTM では、各タスクの情報を管理するために 16 バイトで構成される TCB をタスクごとに設け、タスクの状態管理等を行っている。図 3 に IRTM の TCB の構造を示す。

また IRTM のスケジューリングは、優先順位方式によって行っている。TCB はタスクが生成されるごとに作られるが、生成された順にタスク番号が割り当てられ、タスク番号が小さいほど優先順位が高くなる。

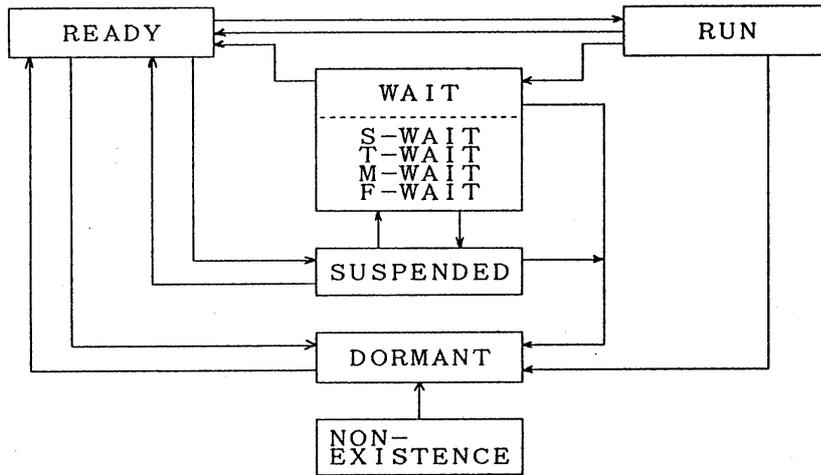


図 2 : IRTMにおけるタスクの状態遷移

| | | |
|-----|---------------------|--|
| 0 | タスク・ステータス 未使用 | → タスクの現在の状態 |
| +2 | スタック・ポインタ // | → タスクの現在のスタック・ポインタの値 (各レジスタは、スタック上に格納される) |
| +4 | メールボックス // | → 受信したメッセージの先頭番地 |
| +6 | イベント・フラグ 待ちパターン | → フラグ待ちタスクの要求パターン |
| +7 | イベント・フラグ | → イベント・フラグ |
| +8 | 時間経過待ちタスク へのポインタ | → 時間経過待ち行列の次のリンクを示すポインタ |
| +10 | 待 ち 時 間 | → 待ち時間 (前にリンクされているタスクとの 待ち時間の差) |
| +12 | スタック・ポインタの 初期値 | → タスクが利用できるスタックのBOTTOM値 |
| | プログラム・カウンタの 初期値 | → タスクの実行開始番地 |

図 3 : IRTMにおけるTCBの構造

4. IRTMのシステム・コール

システム・コールは、ユーザが開発したタスクの中で IRTM が持つ機能を利用するためのインターフェースである。ユーザは、タスクの中で適当なシステム・コールを発行することによって、IRTM が提供するリアルタイム・マルチタスク環境を利用することになる。IRTM では、表1に示す16種類のシステム・コールと4種類のマクロ定義を用意した。

表1：IRTMのシステム・コールの種類と概要

| システム・コールの番号 | システム・コール名 | 機能の概要 |
|-------------|-----------|--------------------------|
| 0 | CRE?TSK | TCBを生成する |
| 1 | STA?TSK | 他タスクを開始する |
| 4 | EXT?TSK | 自タスクを終了する |
| 7 | TER?TSK | 他タスクを強制終了 |
| 12 | SUS?TSK | 他タスクを強制中断する |
| 13 | RSM?TSK | 他タスクの強制中断を解除する |
| 14 | SLP?TSK | 自タスクを待ち状態に移行させる |
| 15 | WAI?TSK | 自タスクを時間経過待ち状態に移行させる |
| 16 | WUP?TSK | 他タスクの待ち状態を解除する |
| 21 | DEL?FLG | 自タスクのイベント・フラグをリセットする |
| 22 | SET?FLG | 他タスクのイベント・フラグをセットする |
| 23 | WAI?FLG | 自タスクをイベント・フラグ待ち状態に移行させる |
| 32 | SND?MSG | メッセージを送信する |
| 33 | RCV?MSG | メッセージを受信する |
| 48 | SET?TIM | システム時計の値をセットする |
| 49 | GET?TIM | システム時計の値を得る |
| | ENTQINT | 割込みハンドラを開始する |
| | ENTQSCH | 割込みハンドラを終了後スケジューラを起動する |
| | RETQINT | 割込みハンドラをスケジューラを起動せずに終了する |
| | MONQINT | IRTMの割込みハンドラを実行する |

5. IRTM用デバッガの概要

IRTM用のデバッガとして、従来から開発を行ってきたシングルタスク用のリモート・モニタを搭載した。

IRTMはマルチタスクOSであるため、複数のタスクが同時に存在し、動作している。このような状況下でリモート・モニタをデバッグとして利用できるよう、次のような機能を追加した。

デバッグ対象のタスクを変更する機能

システム・コールを実行する機能

タスクの状態を表示する機能

このリモート・モニタは、IRTM上の一つのタスクとして動作する。タスクの優先順位は最も高く設定されており、タスク番号は、0である。

またさらに、デフォルトのデバッグ・タスクとして、タスク番号1のタスクが同時に登録され、通常のシングルタスクのデバッグとしてのリモート・モニタと同様に、タスクが完成する前の、アルゴリズムの確認などに利用できるよう配慮した。

6. 結 言

生産管理システムの生産現場監視用コンピュータに必要な、リアルタイム・モニタを開発した。開発したリアルタイム・モニタは、タスクを6つの状態で管理する。またユーザが利用できる機能は、16種類のシステム・コールと4種類のマクロ定義である。モニタのサイズは、約1KBであった。さらに、リモート・モニタをデバッグとして搭載した。

これをもとに平成2年度は、ホスト・コンピュータと生産現場監視用コンピュータとの通信システム、およびホスト・コンピュータ、現場監視用コンピュータのソフトウェアを開発し、システムの試作を行う予定である。

参考文献

- 1) 「解析マルチタスク」 佐々木 茂 著 CQ出版社
- 2) 「I TRON 入門」 坂村 健 編 岩波書店
- 3) 「RX136 16ビット・リアルタイム・オペレーティング・システム 基礎編」
日本電気(株)