

## V60 リアルタイム OS におけるタスク

3V-1

## — 多重仮想空間の導入 —

南沢 英雄、永作 浩之、田中 英信、古城 隆

日本電気(株) マイクロコンピュータ・ソフトウェア開発本部

## 1. はじめに

従来、リアルタイム OS は高速性を重視するあまり、安全性から見た場合問題があった。すなわち、システムと複数のユーザタスクが同一の論理空間上に置かれているため、障害のあるユーザタスクは容易にシステム及び他のユーザタスクを破壊することが可能であった。このため、V60 リアルタイム OS では安全なタスク走行環境を実現するため、メモリ管理方式として多重仮想空間方式を採用している。本稿は現在、開発の進められている V60 用リアルタイム OS に関して、その概要とタスクの管理方法、多重仮想空間の実現方法について報告する。

## 2. OS の概要

V60 リアルタイム OS は 32 ビットマイコンのアーキテクチャを最大限に活用し、スループットを損なわずマルチタスク環境を提供するとともに、ROM ベースの超小型組み込みシステムから小型情報処理システム、フォルトトレラントシステムに至る広範囲な応用分野に対して高い信頼性と生産性を実現するシステムを提供することを目的としている。

本 OS は、リアルタイムカーネル(RTK)、ファイルシステム(RFS)、UNIXスーパーバイザ(UTS) の3つのモジュールに分けられ、必要に応じ、RTK のみ、RTK + RFS、RTK + RFS + UTS の構成をとることができる。RTK は OS の核部分で、上位アプリケーションに高速で安全なタスク走行環境を提供する。RFS は UNIX と互換性を維持しながらリアルタイム及びフォルトトレラントな機能を強化したファイルシステムである。UTS は広いアドレス空間を必要とするような情報処理型タスクを実現するためのスーパーバイザで UNIX と互換性をもつ。

マルチタスクシステムではタスク毎の走行環境が高い独立性をもつこと、タスク間の同期・通信が安全で効率の良いことが重要な点になる。完全なマルチタスク環境を実現するため、RTK では V60 チップ上のメモリ管理ユニット(MMU)により多重仮想空間を生成し、タスク毎に独立した論理空間を与える。タスク切り替えには V60 のコンテキストセーブ/リカバリ専用の命令を用いている。

本 OS では、タスク間の同期・通信の手段としてセマフォやメールボックスのような従来方式の提供だけでなく、Ada 言語のタスク間通信(ランデブー)を RTK が直

接サポートする。アプリケーションの記述言語として C 言語に Ada のランデブー構文を導入した、拡張した言語を提供している。[2]

リアルタイムシステムでは実時間応答性能と同時に耐故障性の機能が要求される。RTK ではフォルトトレラント機能として、ユーザの定義した障害判定ロジックに従い、障害の程度に応じて 4 レベルのシステム再立上げ処理(再開処理)を提供している。

RTK のシステムコールインタフェースは 16 ビット・マイクロプロセッサ V20 及び V30 のリアルタイム OS (RX116 : 仕様は ITRON に準拠)と高級言語レベルで互換性がある。

ファイルシステムと TSS インタフェースは、既存の UNIX インタフェースと互換性を持たせることにより、流通ソフトウェアの利用が可能となる。アプリケーションプログラムは UNIX インタフェースとともにいくつかの RTK 機能を利用することができる。

## 3. タスク

リアルタイムカーネルはマルチタスク環境を提供している。タスクにはタスクの動作環境を規定する幾つかの属性が与えられており、これらの属性を総称してタスクタイプと呼んでいる。タスクには必ず 1 つのタスクタイプが対応し、属性の組み合わせによりリアルタイムタスクから TSS タスクまで幅の広いタスク走行環境を規定する。リアルタイムカーネルのタスクを設計するにあたり次の条件を考慮した。

- 1) リアルタイムタスクと TSS タスクをタスク管理上一元化する。
- 2) リアルタイムタスクの実時間要求を損なうオーバーヘッドは極力避ける。

タスクの属性には、タスクの置かれるメモリ空間の保護機構のレベルを示すレベル値、タスクへのメモリ資源の割り付け時期を規定するタスク生成方式、テキストやデータのロード元を規定する常駐、非常駐の指定、タスクの優先度、タスクに割り当てるタイムスライス値、スケジューリング指定などがある。

タスクの生成方式はタスクが走行するために必要なメモリ資源(管理データ(TCB)、メモリセグメント)をいつ

Task Management on V60 Real Time OS

Multi Virtual Space Implementation

Minamisawa Hideo, Nagasaki Hiroyuki, Tanaka Hidenobu, Kojo Takashi

NEC Corporation

タスクに割り付けるかにより、スタティック型とダイナミック型の2種類のタスクに分けられる。スタティック型タスクはシステム初期設定時、タスクにメモリ資源を割り付けてしまい、タスクの起動・終了時のメモリ割り付け処理を省略する。ダイナミック型タスクはタスク起動時に資源を動的に割り付け、終了時に解放する。スタティック型はダイナミック型に比べタスク起動・終了時のオーバーヘッドがなく資源が常に割り付けられている点を除けばダイナミック型タスクとまったく同じ扱いを受ける。

常駐、非常駐の指定は、タスク起動時のテキスト、データのロード元をメモリに常駐させたロードイメージからロードするのか、ファイルからロードするのかの指定である。

タスク生成方式、常駐、非常駐の指定は、タスク起動時のオーバーヘッドを最小限とし、タスクのリアルタイム性を極力損なわないようにしている。また、タスクとしてリアルタイムタスクだけでなく TSS タスクも含め統一のとれた管理を行なえるようにタスクの属性を決めている。

#### 4. タスクのメモリ空間

従来のリアルタイム OS は高速性を重視するあまり、安全性から見た場合、問題があった。システムと複数のタスクが同一の論理空間上に置かれているため、バグのあるタスクが容易にシステム及び他のタスクを破壊することが可能であった。しかし、マイクロプロセッサが 8 又は 16 ビットの頃は処理能力も低く、機能的にも十分でないため(多重仮想空間方式を高速に実現する機能はない)大型機等で導入されているメモリ管理方式を採用することはできなかった。

V60 リアルタイム OS のメモリ管理方式を選択するにあたり、次の理由により、多重仮想空間方式を採用した。

- 1) 安全なタスク走行環境を実現するためには、強力なメモリ保護機構が必要である。
- 2) 32 ビットマイコン V60 チップ上の MMU で実現される多重仮想空間方式、高速な専用命令(コンテキスト切り替え命令等)、高い処理能力を利用できる。
- 3) 情報処理型の TSS タスクを実現するためには広いアドレス空間が必要となる。
- 4) TSS タスクとリアルタイムタスクをメモリ管理上一元管理したい。

V60 のメモリ保護機構は 4 レベルのリングプロテクションをサポートしている。リアルタイム OS では、レベル 0 にリアルタイムカーネルを、レベル 1 にリアルタイムタスクと TSS スーパーバイザを、レベル 3 に TSS タスクを割り付けている。これにより、RTK は TSS スーパーバイザ及びタスクから、リアルタイムタスク及び TSS ス

ーパーバイザは TSS タスクから保護される。

V60 ではチップ上のメモリ管理ユニット(MMU)により、ページング方式の多重仮想空間を生成し、4 ギガバイトのアドレス空間をタスク毎に与えている。タスクの論理空間のうち、最初の 2 ギガバイトがタスク毎に独立な個別空間に、残りの半分が共通空間になっている。共通空間にはリアルタイムカーネル、常駐のタスクテキスト、共有データ、メモリバッファが割り付けられる。個別空間にはリアルタイムタスクの場合、スタック等のデータエリアが、TSS タスクの場合、テキスト及びデータが割り付けられる。タスクに割り付けられた空間のうち、タスク個別空間はタスク切り替えのたびに切り替えられ、他のタスクからは直接アクセスされず安全なタスク走行環境が実現できる(図 1)。

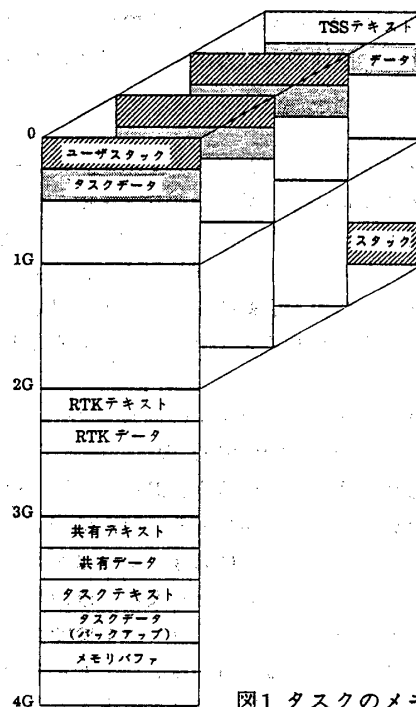


図1 タスクのメモリ空間

#### 6. おわりに

現在開発中の V60 リアルタイム OS に関して、その概要とタスクの管理方法、多重仮想空間の実現方法について報告した。今後は、カーネルの性能評価、OS のチューンアップを行なう予定である。

#### 参考文献:

- [1] 古城、他: V60リアルタイムOS設計、情報処理第 33 回全国大会講演集
- [2] 土屋、他: V60リアルタイムOSにおけるタスク間通信 — ランデブーの実現 —、情報処理第 33 回全国大会講演集