時分割スケジューリング機能を有する時間駆動アーキテクチャ向けリアルタイム OS

伊丹 悠一　横山 孝典　志田 晃一朗　兪 明連
武蔵工業大学

1. はじめに

組み込みシステムでは，時間制約を満足するとともに，複雑な処理を分割して開発効率を上げる目的でリアルタイム OS が使われる．近年では処理量の増加が顕著で，リアルタイム OS の使用が必須になっている．

リアルタイム OS では，タスクのスケジューリングをどのように行うかが重要である．一般にはリアルタイム性の確保のため，優先度ベースのスケジューリングがよく使われる．また，自動車制御など時間制約の厳しいハードリアルタイムシステムでは，制御処理が制御同期（サンプリング周期）に応じて実行されることが多い．このような処理は，その実行周期の変動（ジッタ）が問題になるので，時間の周期にもとづいて処理を行う時間駆動アーキテクチャ[1] が適している．時間駆動アーキテクチャに対応するには，ジッタの少ないスケジューリングが必要になる．時間駆動タスクを扱うリアルタイム OS として，OSEK/VDX によって仕様策定された OSEKtime [2] がある．

ところが，実際の組み込みシステムでは時間駆動タスクのみではなく，入力イベントに応じて実行される非時間駆動タスクも存在する．そこで，時間駆動タスクと非時間駆動タスクが混在したスケジューリングが可能なリアルタイム OS が提案されている．

時間駆動タスクと非時間駆動タスクを同時に動かすため，OSEK OS [3] を OSEKtime のサブシステムとして動作させる方法が OSEK/VDX により提案されている．非時間駆動タスクは OSEK OS 上で動作させることが必要になる．このとき，OSEK OS は時間駆動タスクが動作していない時間を使い，非時間駆動タスクを動かさなければならない．そのため，この方法ではタスクの切り替えに，OSEKtime のディスパッチ処理と OSEK OS のディスパッチ処理の 2 つが必要になり，オーバヘッドが大きい．

また，OSEK OS を単独で動作させ，時間駆動タスクを管理するためのモジュールを設ける方法も提案されている [4]．しかし，この方法では時間駆動タスクも OSEK OS の管理下で動作することになる．一般には時間駆動タスクは非時間駆動タスクよりも優先させて実行することが要求される．そのため，OSEK OS のスケジューラを介さずに時間駆動タスクを実行できることが望ましい．

そこで本研究の目的は，時間駆動タスクと非時間駆動タスクが混在する環境で，効率のよいスケジューリングを提供できるリアルタイム OS を提案，実装することである．

2. 時間駆動タスクと非時間駆動タスクのスケジューリングポリシー

我々は時間駆動タスクと非時間駆動タスクを明確に区別したスケジューリング方式を提案する．具体的には，時間駆動タスクのみを行う時間と非時間駆動タスクのみを行う時間を設け，一定時間ごとに切り替え，交互に動作させる．提案手法のスケジューリング例を図 1 に示す．

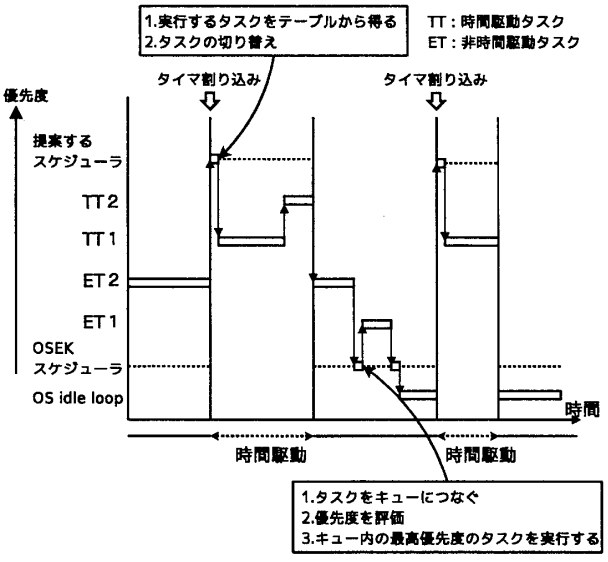

図 1　提案手法のスケジューリング例

この図のように 2 種類のタスクを明確に分けるのは，それぞれのタスクでスケジューリングポリシーを変えるためである．非時間駆動タスクは，一般的な優先度ベースのスケジューリング方式を

A Real-Time Operating System with Time Division Scheduling for Time-Triggered Architecture
Yuichi Itami, Takanori Yokoyama, Koichiro Shida and Yoo, Myungryun,
Musashi Institute of Technology

取る．一方，時間駆動タスクのみを行う時間では，起動するタスクを選択する際にタイムテーブルを参照するようにする．このタイムテーブルには実行周期ごとに時間駆動タスクが登録される．この登録はシステムの構築時に行い，そのときタイムテーブルを作成する．

この方法では，時間駆動タスクについては行っていた優先度の評価を行わず，起動するタスクを決めることができる．これにより，優先度ベースの方式と比べて時間駆動タスクを起動する際のオーバヘッドを減らすことができる．

3. 実装方法

OSEK OS をベースに，前節で提案したスケジューリングを定義するための機構実装する．具体的には，優先順位，実行周期が静的に決められるという時間駆動タスクの性質から，OSEK OS のスケジューラとディスパッチャに変更を行う．対象として，OSEK OS 仕様の実装である TOPPERS/OSEK カーネル [5] を扱う．

3.1 レディキューの操作処理

OSEK OS では優先度ベーススケジューリングを採用しているため，タスクの起床，切り替え要求を出すシステムコールなどを使用すると，タスクはレディキューにつながれる．キューは優先度ごとに用意され，要求を出されたタスクの優先度に合った部分につながれる．この方法はイベントの発生に応じて起床されるタスクには適している．しかし，時間駆動タスクの場合は起動タイミングが分かってるため，この処理はオーバヘッドになる．そこで，時間駆動タスクのみを扱う時間は，この処理を行わずに起床要求があったら即座にタスクを起動する．これにより，スケジューラのオーバヘッドを減らすことができる．

3.2 時間駆動タスクの実行処理

時間駆動タスクを周期的に起動する際に参照するティック値は，ハードウェアタイマの割り込みで供給する．供給されたティック値をもとにソフトウェアタイマを作成し，時間駆動タスクの動作周期かを判定する．もし動作周期であった場合は，その時間駆動タスクを起床する．

OSEK OS は割り込みを 2 種類定義しており，サービスコールが使用できる ISR カテゴリ 1 と使用できない ISR カテゴリ 2 がある．ティック値を供給するハードウェアタイマ割り込みは ISR カテゴリ 1 を使用する．これにより，時間駆動タスクは他のタスクや割り込みよりも高い優先度で実行されることになる．図 2 に提案するリアルタイム OS のプロセッシングレベルを示す．

ただし，ISR カテゴリ 1 を使用した場合，サービスコールでタスクを起動できない．そのため，時間駆動タスクの起床を ISR カテゴリ 1 でも実行できるサービスコールを新たに追加した．

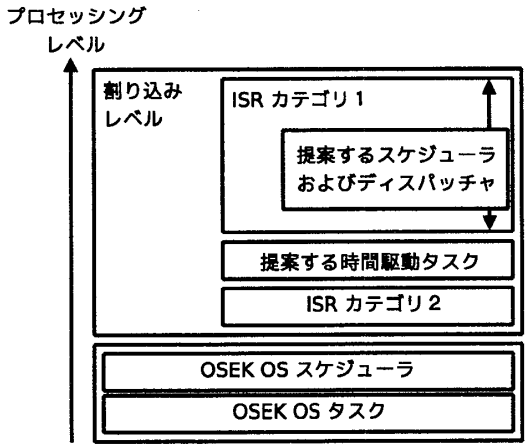

図 2 提案手法のプロセッシングレベル

4. おわりに

時間駆動アーキテクチャに対応可能なリアルタイム OS の提案を行い，それを実現するためにスケジューラとディスパッチャのプロトタイプを作成した．現在は，作成したプロトタイプの性能評価を行っている．

今後の課題として，提案したリアルタイム OS に対応したシステムジェネレータ [6] の開発を予定している．また，OSEKtime で定義されている，時間駆動タスク処理中にあった割り込みの受付だけを行う機能などの追加を検討している．

参考文献

[1] H.Kopetz, Should Responsive Systems be Event-Triggered or Time-Triggered? IEICE Trans. Inf. & Syst., Vol.E76-D, no.11 pp.1325-1332, 1993.

[2] OSEK/VDX, OSEK/VDX Time-Triggered Operating System Version 1.0, OSEK/VDX, July 24th, 2001

[3] OSEK/VDX, OSEK/VDX Operating System Version 2.2.3 , OSEK/VDX, February 17th, 2005

[4] 服部博行, 大西秀一, 森川聡久, 片岡歩, 中村俊夫, 高田広章 ; オープンソース FlexRay 通信 : TimeTriggered OS と FlexRay 通信ミドルウェア.

[5] TOPPERS/OSEK カーネル ; http://toppers.jp/osek-os.html

[6] OSEK/VDX, OSEK/VDX System Generation OIL: OSEK Implementation Language Version 2.5 , July 1, 2004