マルチコアプロセッサ環境における組込みシステム向け VMM の研究

前田剛志1　　並木美太郎 1

1.　はじめに

近年，ハードウェアの高度化に伴い組込みシステムへの要求も高度化しており，特に情報システムとの連携を行うことによる，高度な組込みシステムを構築することが求められるようになった．

このような要求にこたえるために，仮想化技術を用いて RTOS と汎用 OS の共存を実現する実行基盤の研究[1]が存在する．これには仮想化処理のオーバヘッドによるリアルタイム性の阻害，予測可能性の損失という課題がある．

本稿では,CPU，メモリ，I/O の仮想化を実現する，組込みシステム向け仮想マシンモニタ MVM の設計および一部機能の実装を行った．実装に用いたターゲットボードは Zynq-7000 SoC ZC702 であり，これは組込みシステムで広く利用される ARM プロセッサを内蔵しているうえ，FPGA と連携することができる．将来的には FPGA の仮想化を実現することを考えている．

2.　課題

前章で述べた実行基盤を実現するため，VMM を利用して，VM としてリアルタイム処理を行う RTOS ，IT システムの処理を行う汎用 OS という構成が考えられる．ただし，VMM を利用する方式の問題として，VM ごとにリアルタイム性が異なり，汎用 OS の処理が RTOS の処理を阻害するといったことが生じる可能性がある．

本研究では，組込みシステムへの要求の高度化への対応，特にリアルタイム処理と IT システムの連携という要求を満たすことを念頭に置いている．そこで，以下のような課題が挙げられる．

- リアルタイム要求の異なる VM に対するリアルタイム性の実現

3.　ARM の仮想化機能

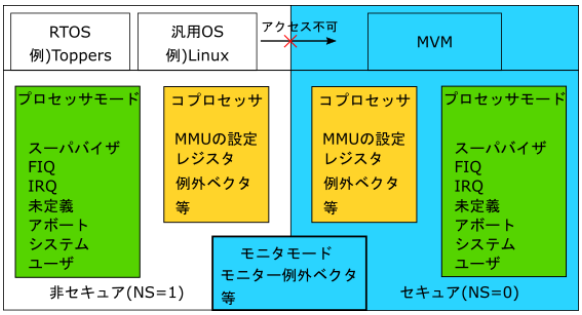

図 1　TrustZone を用いた構成

本研究では，VMM を実現するために，ARM のセキュリティ拡張機能 TrusZone を利用する．

TrustZone はメモリ空間をセキュアワールドメモリ空間とノーマルワールドメモリ空間に分離する．セキュアワールドにはプロセッサが Secure な時にアクセス可能で，プロセッサが NonSecure な状態の時にはアクセスができない．TrustZone の機能を用いることで，各ワールドに独立した例外ハンドラやシステム制御レジスタ，アドレス変換テーブル等を持たせることが可能である．これにより，セキュアワールドが特権を持つが，各ワールドにおいて別なシステムを動作させることが可能である．また，ノーマルワールドからセキュアワールドへの遷移にはモニタモードというプロセッサモードが存在し，ソフトウェアの smc 命令，外部割込みに起因して遷移を発生させることが可能である．

4.　設計

二章の課題を解決するため，本研究にて Type1 仮想マシンモニタ MVM を提案する．全体の構成としては，TrustZone を利用し，MVM 本体をセキュアワールドにて動作させ，ゲスト OS をノーマルワールドにて動作させる．この構成と TrustZone の関係性は図 1 のようになる．

4.1　リアルタイムレベル

MVM は二章にて述べた「リアルタイム性の異な

1 東京農工大学
Tokyo University of Agriculture and Technology

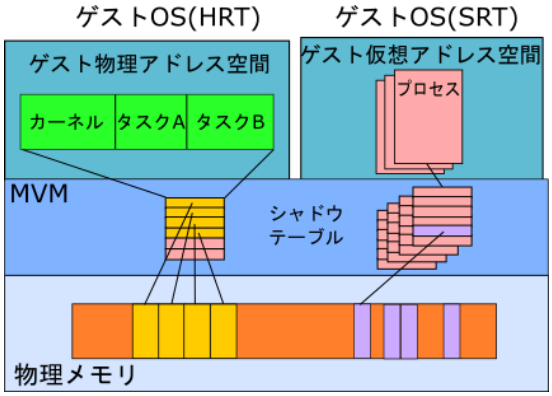

図 2　メモリ仮想化手法

る VM に対するリアルタイム性の実現」の課題を満たすために，リアルタイム性に応じた各ハードウェア資源の仮想化方式を提供する．そこで，以下に示す二段階のリアルタイムレベルをゲスト OS に設定可能にする．

- HRT(ハードリアルタイムゲスト向け)
  他のシステムから独立してリアルタイム性を保証する手法を用いる．
- SRT(ソフトリアルタイムゲスト向け)
  優先的に物理資源を獲得するが，余剰資源は共有する．

ゲスト OS に設定したリアルタイムレベルに応じた仮想化方式で MVM が資源を割り当てることで，異なるゲスト OS がそれぞれ要求するリアルタイム性を保証する．

4.2　CPU 仮想化手法

CPU を仮想化し，よりリアルタイム性の要求されるものに対し CPU の占有率を高めることでリアルタイム要求の異なるゲスト OS を動作させる．リアルタイムレベルの考えに則り，以下の二つの手法を提供する．

- リアルタイムスケジューリング機能
- コア占有機能

コア占有機能では，デッドラインを超えると致命的な事態を起こしうるゲスト OS に対し，プロセッサコアを占有させることで，CPU 仮想化オーバヘッドを回避する．

リアルタイム CPU スケジューリング機能では，複数のゲスト OS を EDF スケジューリングに則り占有されていない CPU を仮想化することで，優先度の高いゲスト OS が優先実行される．

4.3　メモリ仮想化手法

メモリ仮想化に関しても，CPU の仮想化と同様に二つの手法を提供することにより，異なるリアルタイム要求のゲスト OS に対応する．

- ダイレクトマップ
- シャドウページング

厳しいリアルタイム要求のゲスト OS にはダイレクトマップによる仮想メモリを提供する．これは物理アドレス＝仮想アドレスの対応の記述されたページテーブルをゲスト OS 起動時に用意することで，リアルタイム性を実現することを考えている．また，ページサイズの粒度を荒くすることで，TLB ミスの起こる頻度を小さくすることを実現する．これにより，次のシャドウページングよりも低オーバヘッドを実現することができる．

シャドウページングでは MVM 上にゲスト仮想アドレスと実アドレスの対応を記述したシャドウページテーブルを用意し，これを利用することでメモリの仮想化を実現する．図 2 にこれらを図示する．

4.4　I/O 仮想化手法

リアルタイム性を実現する必要があるため，I/O をスケジューリングするといった仮想化処理は，オーバヘッドがシステムに致命的な事態を引き起こす可能性がある．そのため，MVM ではゲスト OS の I/O 要求をハードウェアにパススルーすることにする．

5.　現状

現状として，MVM において，ノーマルワールド側でのゲスト OS の実行およびハイパバイザーコールを含む例外ハンドラの実装が完了しており，ノーマルワールド側のゲスト OS として，ITRON 仕様 RTOS の TOPPERS が動作している．

6.　今後の予定

各仮想化手法に関して，まだ実装が完了していない部分について実装を完了し，複数ゲスト OS 実行時の評価を行う．また，実装環境の ZYNQ の特性である，FPGA 部に関して MVM により仮想化を行い，ゲスト OS に機能として提供することを考えている．

参　考　文　献

[1]　"汎用 OS と専用 OS を高効率に相互補完するナノカーネルの提案と実現", 新井利明, 関口知己, 佐藤雅英, 木村信二, 大島訓, 吉澤康文 情報処理学会論文誌 Vol.46 No.10 pp.2492-2504, 2005-10-15