

組込みシステム学習環境を対象とした 仮想マシンマネージャの開発

青山誠一[†] 早川栄一[‡]

拓殖大学 大学院 工学研究科[†] 拓殖大学 工学部 情報工学科[‡]

seiichi.aoyama@os.cs.takushoku-u.ac.jp[†] hayakawa@cs.takushoku-u.ac.jp[‡]

1. はじめに

近年、組込みシステムは多様化し、ネットワーク接続や外部ストレージなどは必須の機能となっている。このような機能を教育用 OS にも追加し、より学習を支援するために利用したいという要求があるが、これは OS の複雑化を招き、教育用 OS にあるべき単純で理解しやすい設計をとれなくなってしまう。単純な設計を保つためには OS の基本的な機能を提供する部分とデバイス管理を行う部分とで分ける必要がある。

それを実現する手段として、仮想化技術を用いて複数の OS を同時に動作させる方法がある。仮想化の構成として、VMware^[1]のようなハードウェアの完全な仮想化を提供するものと、Xen^[2]などで用いられている準仮想化と呼ばれる方法により、エミュレーションのオーバーヘッドを抑えるものがある。前者の場合、互換性は高いが実行コストが大きく、後者は対象となる仮想環境に対応させるための修正が必要だが実行コストが抑えられる。

組込み用途への適用を考察すると、CPU 速度やメモリ資源が制限されるため、完全な仮想化を提供する方法は向かない。よって準仮想化のようなオーバーヘッドを最小限に抑えるなど、組込みに適した仮想マシンマネージャ(以下、VMM)の設計が必要となる。

そこで本研究では、組込みシステム学習環境「港」^[3]で用いられている教育用ロボットを対象とした VMM を開発し、その教育用 OS に単純な設計と柔軟な拡張性を与え、さらにゲスト OS のデバッグを支援する機能を組み込むことで、学習しやすい環境の構築を目的とする。

2. 方針

組込みシステム学習環境「港」の教育用ロボットに搭載されている SH-3 CPU ボードを対象として VMM の開発を行う。ロボット制御を行う教育用 OS とデバイス管理を行う汎用 OS に分けることで、教育用 OS の構造を複雑化させずに、様々なデバイスのサポートや機能の追加を実現する。また、VMM にモジュールとしてデバッグ機能を用意し、必要に応じて追加することで単体でのデバッグでは難しい割り込み禁止時のログ取得などを支援し、学習環境との連動によって、学習者に情報を分かりやすく提供する。

3. 特徴

オーバーヘッドをできるだけ抑え、応答性を高めるための設計として次に挙げる特徴を持つ。

(1) デバイス管理 OS は特権モードで動作

デバイス管理 OS を VMM と同じ特権モードで動作させることによって単体の場合と同じ動作速度を実現し、デバイス管理における実行コストを抑えることが可能となる。

一般的な仮想化の構成では VMM を特権モード、ゲスト OS をユーザモードで動作させている。そのゲスト OS が特権命令を実行しようとしたとき、VMM がそれを捕捉し命令のエミュレーションを行うことによって、ユーザモードでの動作を可能としている。しかし、このエミュレーションは非常に大きな実行コストとなってしまうため、実行速度の低下を招き、応答性を高める設計の障害となる。

(2) 2 種類のデバイスアクセス手段の提供

各種センサなど単純でリアルタイム性が求められるデバイスは VMM がアクセスを行う仮想デバイス I/O、無線 LAN など制御が複雑でリアルタイム性がそれほど求められない場合はデバイス管理 OS がアクセスを行うポート通信という 2 種類のアクセス手段を提供する。

必要に応じたアクセス手段を選択することで、ロボット制御におけるオーバーヘッドを軽減することが可能となる。

(3) 独立したタイマを利用

動作対象としている CPU は SH-3 ファミリの SH7709S であり、これには三つの独立したタイマを内蔵している。この三つのタイマをそれぞれ VMM、デバイス管理 OS、ゲスト OS に割り当てる。

これにより VMM はタイマのエミュレーションを行う必要がなくなり、そのオーバーヘッドもなくなる。

(4) 機能拡張性

VMM は必要に応じて機能を選択できるモジュール構成とすることで、新たな要求による機能の拡張を容易に実現し、例えば、デバッグ機能が不必要な場合はその機能を外すことで余分なオーバーヘッドを抑えた構成をとることが可能となる。

4. 全体構成

本システムは構成図を図 1 に示す。大きく VMM、デバイス管理 OS、ゲスト OS の三つの部分に分けられる。

特権モードでは VMM とデバイス管理 OS、ユーザモードではゲスト OS が動作し、VMM 内のコア部分が仮想化の基本的な機能を提供し、モジュールとしてデバイス管理 OS やデバッグ機能が用意されている。

Development of a Virtual Machine Manager for Embedded System Learning Environment

Seiichi Aoyama[†], Eiichi Hayakawa[‡]

[†]Graduate School of Engineering, Takushoku University

[‡]Faculty of Engineering, Takushoku University

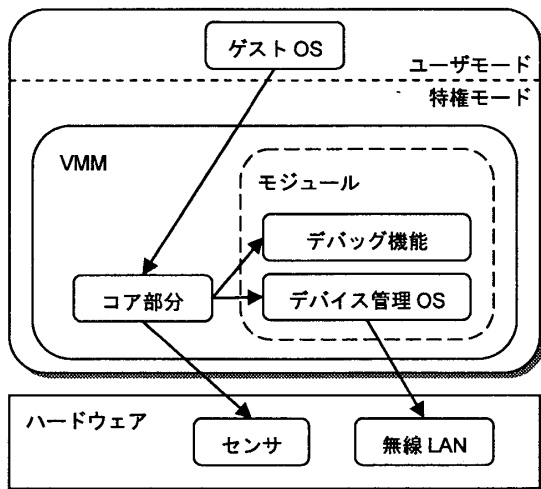


図1 システムの全体構成

5. 設計

5.1 VMM

VMM は次に挙げる機能を提供する。

- (1) 特権命令の捕捉・エミュレーション
特権命令を捕捉し、エミュレーションすることでユーザーモードでのゲスト OS 動作を可能とする。
- (2) ハイパバイザコール

ハイパバイザコール (以下、HVC) とはゲスト OS から VMM に対するシステムコールである。VMM の制御、ポート通信、仮想デバイス I/O を行う命令が用意される。

- (3) 仮想デバイス

ロボット制御用のデバイスを仮想デバイスとして定義する。光センサ、モータ、LED などのロボット制御に不可欠なデバイスを VMM が仮想的に用意し、ゲスト OS からは用意された命令を通してアクセスを行う。こうすることで通常のアクセス命令で生じてしまうハードウェアアクセスのエミュレーション処理が不要となり、オーバーヘッドを小さくすることが可能となる。

仮想デバイスへのアクセスは HVC による仮想デバイス I/O 命令で行われ、それが呼び出されると VMM が代わりに対応する実デバイスへアクセスを行う。

- (4) デバッグ機能

本機能はモジュールとして VMM に組み込まれ、ゲスト OS のデバッグに必要なメモリやレジスタ、割り込み禁止時の状態変化などを監視する機能を提供する。

ゲスト OS の状態が変化する各種イベントに対して VMM は割り込みという形で用意し、対応する割り込みハンドラを登録することで、各状態変化に対応した処理を行うことができる。これによりデバッグ機能の実装が可能となる。

5.2 デバイス管理 OS

デバイス管理を行う OS として Linux を用いる。Linux には豊富なデバイスと機能をサポートしているため、これを利用することで開発コストを抑えることができる。

Linux は VMM と協調させるため、次の部分を変更する。

- (1) 例外ハンドラ

ユーザーモードからの特権命令の実行、およびタイマの割り込みや HVC の呼出しが起こった場合、例外が発生する。CPU に登録されている例外ハンドラは Linux のものであるため、ゲスト OS が特権命令を実行した場合などは VMM の例外ハンドラに処理を移す必要がある。

- (2) 割り込みハンドラ

ハンドラ自身は Linux のものを利用するため、VMM やゲスト OS への割り込みの場合、処理を移すように変更する。

- (3) 使用メモリサイズ

VMM とゲスト OS にもメモリを割り当てるため、Linux が使用するメモリサイズを変更する。

5.3 モジュール構成

モジュール構成は VMM にデバイス管理 OS やデバッグ機能などを追加する方法である。各モジュールは表 1 に示すような統一されたインターフェースを用意し、これらエントリ関数を VMM に登録することで VMM への機能拡張を容易に行うことができる。

表 1 モジュールが提供するエントリ関数

名前	内容
module_init	モジュール初期化処理を行う
module_exit	モジュール終了処理を行う
module_open	処理開始前に呼び出す
module_close	処理終了後に呼び出す
module_read	モジュールからの入力を行う
module_write	モジュールへの出力を行う
module_ioctl	モジュール固有の操作を行う

および VMM は表 2 に示す関数をモジュールに対して提供する。

表 2 VMM がモジュールに提供する関数

名前	内容
request_irq	割り込みハンドラを登録する
free_irq	割り込みハンドラの登録を解除する
disable_irq	指定された割り込みを無効にする
enable_irq	指定された割り込みを有効にする

また、デバッグ機能モジュールの場合は VMM からの割り込み処理を行う必要がある。そこで module_open 関数が呼び出されたときに VMM が提供する request_irq 関数で割り込みハンドラを登録することによって、デバッグ機能モジュールが割り込みを受け取れるようになり、各状態の監視などの機能を実現できる。

6. おわりに

本研究では、組込みシステム学習環境「港」の教育用ロボットを対象とした VMM を設計し、教育用 OS に柔軟な拡張性と単純な設計の両立を達成することを可能とした。

今後の課題は、学習環境との連携をさらに深めて、学習者にとってより理解しやすい環境を構築することである。

参考文献

- [1] VMware
<http://www.vmware.com/>
- [2] Xen
<http://www.cl.cam.ac.uk/research/srg/netos/xen/>
- [3] 「港」プロジェクト
<http://edo.cs.takushoku-u.ac.jp/minato/>