

複数 OS におけるハードウェア資源の割当て方法

中島 雄作[†] 田淵 正樹[†] 榎本 圭[†] 乃村 能成[‡] 谷口 秀夫[‡]

[†]株式会社 NTT データ技術開発本部

[‡]九州大学大学院システム情報科学研究院

1. はじめに

近年, Windows や Linux 等, ユーザが利用するオペレーティングシステム(以降, OS)の選択肢が増え, それぞれの OS の特徴を活かしたアプリケーション(以降, AP)を1つのパーソナルコンピュータ(以降, PC)上で同時に動かしたいというニーズがある. また, 複数 OS 共存環境において, ある OS 環境における障害が他の OS 環境に影響を及ぼすことを避けるために, 各 OS の独立性を高めたいというニーズがある. さらに, 複数 OS 共存環境で周辺装置を利用する際, 単独 OS 利用時と変わらない機能や性能を享受したいというニーズもある.

複数 OS 共存技術の代表的なものとして, 仮想計算機(Virtual Machine)がある. これは主 OS の上で複数の従 OS を稼働させる機構であるため, 以下の問題がある.

- (1) 主 OS の停止が計算機全体のシステムダウンにつながる.
- (2) 主 OS 提供機能により, 従 OS の機能が制約を受ける.
- (3) 主 OS がエミュレートするため, 従 OS における処理速度が低下する.

そこで, 我々はハードウェア資源を各 OS に直接割り当てることにより, 複数の OS を独立に, かつハードウェア性能を十分に活かした状態で, 同一計算機上で時分割走行させる技術として, 複数実計算機(MRM: Multiple Real Machines)を提案した^[1]. 本稿では, OS 相互間の共有部分を極力少なくするための OS 構成法, OS 間の切替え方式及びハードウェア資源の割当て方法について述べる.

2. 方針

仮想計算機の諸問題を解決し, 複数の OS が互

Hardware resource allocation on multiple operating systems
Yusaku NAKAJIMA[†], Masaki TABUCHI[†],
Kei MASUMOTO[†], Yoshinari NOMURA[‡] and
Hideo TANIGUCHI[‡]

[†]Research and Development Headquarters, NTT DATA Co.

[‡]Faculty of Information Science and Electrical Engineering,
Kyushu University

いに影響を及ぼさないようにするため, MRM は以下の方針に基づいて構成する.

- (1) OS 相互の影響を排除するため, OS 間での資源やプログラムの共有は行わない.
- (2) 市販 OS を利用可能にするため, 共存させる OS のソースコードの改造を行わない.
- (3) 各 OS の性能の低下を避けるため, エミュレートによる資源の共有を行わない.
- (4) 外部割込みの応答性を高めるため, 可能な限り速く, 該当の割込み処理ルーチンを実行できるようにする.

3. ハードウェア資源の割当て方法

前記方針を満たすため, MRM では, 各ハードウェア資源をそれぞれ一つの OS に専有させることで各 OS の独立性を確保する. 但し, PC のハードウェア機構上, CPU やタイマ等, PC 内に一つしかなく, 全ての OS が利用必須な資源については共有が必要となる.

各 OS が利用し, 制御すべきハードウェア資源として, 以下のものがある.

- (1) CPU 及びタイマ
- (2) メモリ
- (3) 周辺装置

(1)については, PC 内で唯一であるため, 各 OS で共有せざるを得ない. CPU については時分割で各 OS に割り当てる. また, 提案する OS 切替え方式では, タイマからの割込みを契機に OS を切替えるので, 各 OS がタイマを検出することが可能となる.

(2)については, 各 OS 毎に物理メモリを領域分割して利用する.

(3)については, 各装置は個別の OS に排他的に割り当てられ, 専有した OS が管理を行う. 周辺装置からの割込みが発生した場合には, 割込み発生時に CPU を利用している OS が割込みを受け, 当該周辺装置を管理する OS に切替え, 割込み処理を実行する.

本稿では, (1)及び(2)について詳細に説明する.

3.1 CPU

CPU を時分割で OS に割り当てることにより、OS 環境が切替わることになる。マルチタスク OS 上のタスクスケジューリングと同様に、ある微小な単位時間においては一つの OS が CPU を専有して利用する。MRM では、特定の OS が長時間にわたり CPU を独占することを避けるため、一定時間を経過したときに発生するタイマ割り込みや周辺装置からの割り込みを契機に、走行する OS を切替える。これら割り込みを契機に切替えることにより、以下のような利点を持つ。

- (1) 割り込み処理の最初に AP の状態を退避するため、AP の走行状態に関係無く OS 切替えが可能。
- (2) OS 切替え時のプログラムカウンタが割り込み処理ルーチンの中にあることが保障されるため、簡単な工夫によりプログラムカウンタの退避・復元が不要。

MRM では、OS 切替えの前後でプログラムカウンタの値が連続するように割り込み処理ルーチンを配置することにより、ページマップテーブルと割り込み管理テーブル等を切替えるだけで、OS 切替えを実現している。

例として、OS2 走行中に、OS1 が管理する周辺装置の割り込みが発生した場合の動作及び、OS1 走行中に、OS2 が管理する周辺装置の割り込みが発生した場合の動作を、図 1 に基づいて説明する。

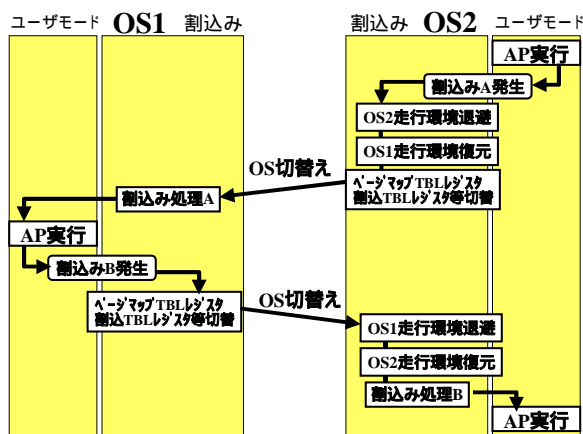


図 1 OS 切替え方式

- (1) OS2 の AP 実行中に、OS1 の管理する周辺装置の割り込みが発生し、OS2 の割り込み処理が呼び出される。
- (2) OS2 走行環境を退避し、前回退避されていた OS1 走行環境に復元する。
- (3) OS2 から OS1 に切替えるため、ページマップテーブルレジスタと割り込み管理テーブルレジスタ等を OS1 用のものに切替える。
- (4) OS1 では当該周辺装置の割り込み処理 A を実行した後、割り込み復帰してユーザーモードに戻る。

- (5) OS1 の AP 実行中に、OS2 の管理する周辺装置の割り込みが発生し、OS1 の割り込み処理が呼び出される。
- (6) OS1 から OS2 に切替えるため、ページマップテーブルレジスタと割り込み管理テーブルレジスタ等を OS2 用のものに切替える。
- (7) OS1 走行環境を退避し、前回退避されていた OS2 走行環境に復元する。
- (8) OS2 では当該周辺装置の割り込み処理 B を実行した後、割り込み復帰してユーザーモードに戻る。ここで、各割り込み処理ルーチンの OS を切替える命令のアドレスを、両 OS の仮想メモリ空間上で整合させる必要がある。また、OS2 側で双方の OS 走行環境（レジスタ）を退避・復元しているのは、OS1 を市販 OS にすることを想定しており、極力変更せずに使用するためである。

3.2 メモリ

図 2 に示すように、各 OS 毎に物理メモリの使用範囲を割り当てることで、OS 間で非共有な状態を構成する。多くの OS では、OS 起動時に BIOS から物理メモリ情報を取得し、メモリマップを作成し、それに基づいて自 OS の物理メモリ使用領域を決定する。MRM では、最初に起動した OS が他 OS 利用分も含めてメモリマップを計画し、自 OS で使用する物理メモリ領域を設定する。そして後から起動する OS の初期化ルーチンを監視し、BIOS から物理メモリ情報を取得する命令を、当該 OS に割り当てられた物理メモリ領域を使用するような命令に、実行直前に置き換えることで、図 2 のようなメモリ構成を実現する。

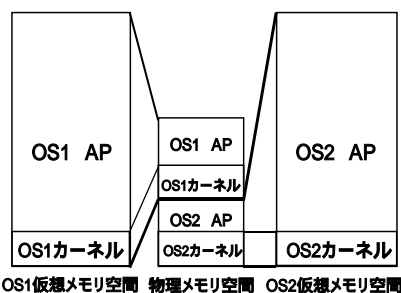


図 2 メモリ構成図

4. おわりに

今後は、本方式における OS 切替えのオーバーヘッド等の性能評価を行う予定である。

参考文献

- [1] 谷口秀夫, 乃村能成, 田中一男, 大塚作一, 井上友二: ハードウェアを非共有する複数オペレーティングシステムの構成法, 情処研報, Vol.2002, No.79, pp.47-54, 2002.