

ハードウェア仮想化による複数 OS の並列実行

腰越 修

近年、仮想化技術を利用してマルチタスクより下のレイヤである OS そのものを複数実行してタスクの独立性を高めるといった手法が普及しつつある。本稿では、組み込みシステムにおいてハードウェアを仮想化し、複数の OS を同時に動作させる手法について解説する。
(編集部)

1. 複数の OS を同時に動作させる仮想化技術

ハードウェアを仮想化して複数の OS を同一のハードウェアで動作させるという技術そのものは新しいわけではありません。メインフレーム(汎用大型コンピュータ)においては1960年代よりIBM社が先駆的に研究と開発を進めてきました。今日のサーバ分野ではVMwareやXenによるサーバの仮想化が一般化しており、世界中の多くのサーバで使用されています。

ここではサーバ分野ではなく、組み込み分野の仮想化技術について、その利便性と仮想化を実現する仕組みを解説します。仮想化ソフトウェアについては、VirtualLogix社のVLXを例に説明します。サーバ分野と異なり、組み込み分野にはCPUやメモリといったハードウェア資源の制約、またリアルタイムOSのリアルタイム性の確保という要求があるので、サーバ分野の仮想化ソフトウェアをそのまま使うことは現実的ではありません。

組み込み分野では、仮想化することによるハードウェア資源の消費を最小限にとどめる必要があります。つまり、1)小さいメモリ・フットプリントで、2)わずかなCPUオー

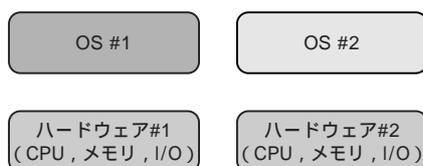
バヘッドで、3)素早い動作を実現する、ということです。

仮想化という言葉はさまざまな意味に用いられますが、ここではハードウェアを仮想的に複数に見せ、結果として複数の OS を単一ハードウェア・セット上で動作させることとします。組み込みシステムのアーキテクチャは、通常、一つのCPUに対して一つのOSです。このCPUとメモリ、各I/Oを仮想的に複数に見せることにより、同一(単一)ハードウェア上で複数OSを動作させます。

図1を見ると、あたかも仮想化レイヤがその上のOSたちを支配しているように見えますが、実際には、ごくシンプルな仕事しか行っていません。各OS(ゲストOSと呼ぶ)に対し、1)優先順位ごとあるいは時間ごとにCPU資源を割り当て、2)各ゲストOSがアクセスできるメモリ領域をパーティションし、3)アクセスできるI/Oデバイスを割り当て、4)ゲストOS間の通信機構を提供している程度のことしかしていません。

仮想化レイヤがでしゃばることなく、わずかなCPU資源の消費とわずかなメモリ使用量で、できるだけゲストOSが本来持つ(仮想化前の)ネイティブな特性をそのまま生かすような形で仮想化を実現しています。これにより、ハードウェア資源の制限の厳しい組み込み機器上でも実用的に利用できるものとなっています。

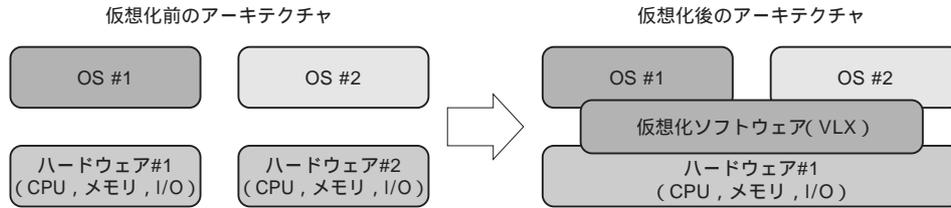
仮想化前のアーキテクチャ



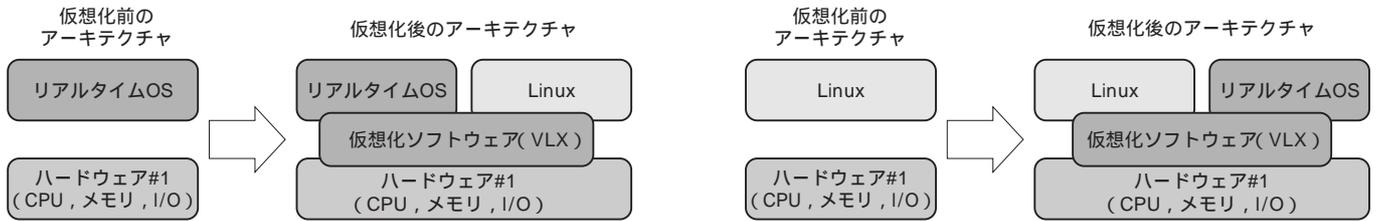
仮想化後のアーキテクチャ



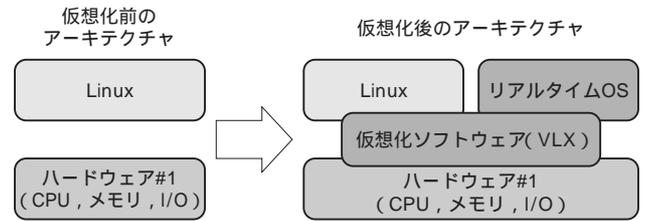
図1
仮想化レイヤを導入することにより、ハードウェアを一つに統合する例



(a) CPU統合によりハードウェアのコストを削減する例



(b) 新規にLinuxを追加する例



(c) 新規にリアルタイムOSを追加する例

図2 複数のOSを走らせる例

2. OSを複数動作させるメリット

では組み込み機器上で複数のOSを走らせるメリットは何でしょうか。このことについて、以下に示します。

CPU統合によるハードウェア・コストを削減する複数のハードウェア(CPU)に分散していたシステムを単一ハードウェア環境に統合すると、純粋にハードウェアのコストを削減できます[図2(a)]。つまり、複数のOS機能を実現するためにOSの数だけCPUを搭載していた場合、いくつかのOSを一つにまとめることによりCPUの数を減らせます。その場合も無理やり複数のOS(の機能)を統合するのではなく、仮想化レイヤを用いて統合すれば、ネイティブ環境をほぼそのまま仮想化環境に移行できます。

全体のシステムをそのままに新機能を導入

LinuxなどのOSと、安定動作の確認された成熟したOSを共存させることを目的に、複数のOSを利用することがあります[図2(b)]。具体的には、使い慣れた既存OSには手を加えずそのまま実装し、仮想化レイヤにLinuxを追加することにより、Linuxがもつ機能をシステムに追加できます。使い慣れた全体のシステム設計を変えることなく、新機能を追加できます。例えば中核の機能はリアルタイムOSに任せ、Linuxの持つGUIや通信プロトコルを加えることができます。

リアルタイムOSを容易に導入

上の例とは逆に、Linuxなどのシステムを既に構築している場合、その基本設計には手を加えず、リアルタイムOSを追加することでリアルタイム・アプリケーションを実装することが考えられます[図2(c)]。あるいは(誤解を恐れずに言えば)Linuxベース機器の仮想的なリアルタイム化を実現できます。リアルタイムLinuxや高価な商用リアルタイムOSを導入することなく、Linuxの豊富な資産をそのまま利用し、かつリアルタイム性が求められるアプリケーションだけリアルタイムOS上で動作させるわけです。これも使い慣れた全体のシステム設計を変えることなく、欲しい機能だけを手に入れる手段として仮想化を導入する例です。

開発期間と開発コストを削減

動作CPUさえ変わらなければ、仮想化の前と後でアプリケーション・ソフトウェアを変更する必要はありません。そのため、ソフトウェア資産を容易に再利用できます。仮想化ソフトウェアが対応するOSであれば、OSの修正も不要です。既存OSに無理やり新たな機能を追加しようとすると、時間と手間がかかります。それと比較すると、仮想化による統合は開発期間と開発コストを削減できます。

アーキテクチャを共通化して保守の手間を軽減

例えば廉価版の機種をシングルCPUで構築し、高機能機種を2CPU(2OS)で構築する場合、シングルCPU上にも仮想化レイヤを導入することにより、ハイエンド・モデル