

低レベル VM を利用した , ハードウェア仮想化技術の開発 仮想計算機の導入によるハードウェア仮想化技術の開発

1. 背景

計算機が1 人1 台の時代は既に到来しており、強力な計算機を1 人で複数台所有することもはや珍しいことではない。また同様に一昔前には大学の研究機関や会社だけに構築されていたLAN 環境も、安価なNIC やネットワーク機器を利用して、個人レベルで手軽に構築することができるようになり、家庭で高速なLAN 環境を構築する人も爆発的に増えてきている。もはやNICは今どきのコンピュータには標準装備、なくてはならないものである。

このような高速なLAN 環境を手に入れてしまうと、そこには無限大の可能性が生じてくる。今まで常識的に考えられなかったことや、不可能であると思われてきたことも現実味を帯びてきている。私が以下に提案するソフトウェアもその1 つであると確信している。

2. 目的

複数台のマシンを所有し、操っていると、どうしてもあるマシンに装着されているデバイスを、他のマシンでも利用したいという思いに駆られることがある。しかし、ノート型コンピュータや、最新のスリム型コンピュータは、基本的に一度買ってしまうと、内部のハードウェア構成を物理的に変更することは不可能とはいえないまでも、きわめて困難で冒険的な作業である(メーカー保証外になってしまう)。またそういった制約をクリアしているコンピュータであっても、物理的にハードウェア構成を変更することは、一度分解して内部の機器を取り出して差し替えるといった作業をこなさなければならない。こういった作業にはハードウェアを痛めてしまうかもしれないといった懸念があり、なにより専門知識が要求される。また他の物理的制限、例えばスロットの不足、インタフェースの相違、根本的に変更不可能(基板に直接取り付けられている)、といった制限もあり、どうしても不可能な場合に直面することも珍しくない。あきらめて仕方なく新しく対象マシンにあった同様のデバイスを買うという最終手段も考えられるが、これすらも不可能な場合がある。例えば研究用に開発したデバイスで、替えがないといった場合や、その機器が非常に高価なもので、新しく購入するのは予算不足であるといった状況が考えられる。こういった場合には、あきらめてしまうしかほかに道はない。同じIA-32 アーキテクチャで動くデバイスであるのに、相互利用できないというのは大変不便であると考える。

そこで本提案では、IA-32 アーキテクチャ(PC/AT 互換機)を対象とし、ネットワークを介して接続された計算機デバイスを、ローカル側の計算機から自由自在に操る技術を開発することを目的とする。ネットワークを介して利用可能なデバイスを、リモートデバイスと呼ぶ。

3. 開発の内容

本ソフトウェア(以下VM)の概念図を図2に示す。本プロジェクトでは、PC/AT 互換機(IA-32アーキテクチャ)上にて動作するハードウェア仮想化技術を開発した。本ソフトウェアの最大の目標は、ハードウェアをCPU のin, out 命令レベルで仮想化することにある。その

ため、本プログラム上でOS を動作させ、OS から発行されるin, out 命令を確実に監視する方法をとった。このVM 上で動作するOS をゲストOS と呼ぶ。本ソフトウェアはPC/AT 互換機上で動作するPCエミュレータのように動作し、ゲストOS は起動中、VM の存在を意識することは無い。

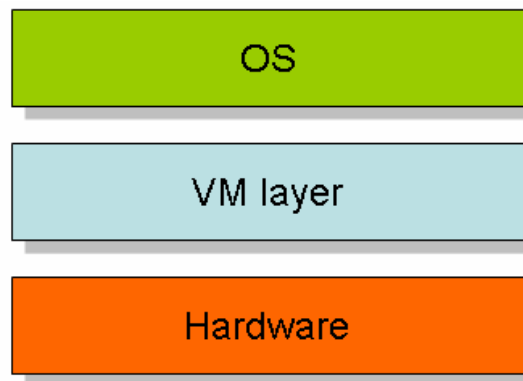


図2 . 概念図

VM は OS に仮想化されたハードウェアを提供する必要がある。そのため、IA-32 アーキテクチャのエミュレーションを提供するためのプログラムを開発した。現時点では完全な CPU のエミュレーションを提供するには至っていないが、ほとんどの CPU 特権命令を実装し終えた。

IA-32 エミュレーションの能力を確認するため、ゲスト OS は DLXLinux と呼ばれる Linux を使用した。DLXLinux とは、一通りの GNU ツールと、Linux のカーネルをまとめた、GNU/Linux システムで、以下のサイトからダウンロードできる。

<http://rpm.pbone.net/index.php3/stat/4/idpl/1259462/com/bochs-dlxlinux-2.0.2-4mdk.i586.rpm.html>

DLXLinux 本体には、再コンパイル等の手を一切加えずに起動させることを目標としたが、完全な起動には至らなかった。しかしながらかなりの部分の実装が終了しており、DLXLinux も完全起動とは行かないものの、一部を起動させることが可能である。図 3 に現時点での DLXLinux の起動状況を示すスクリーンショットを示す。

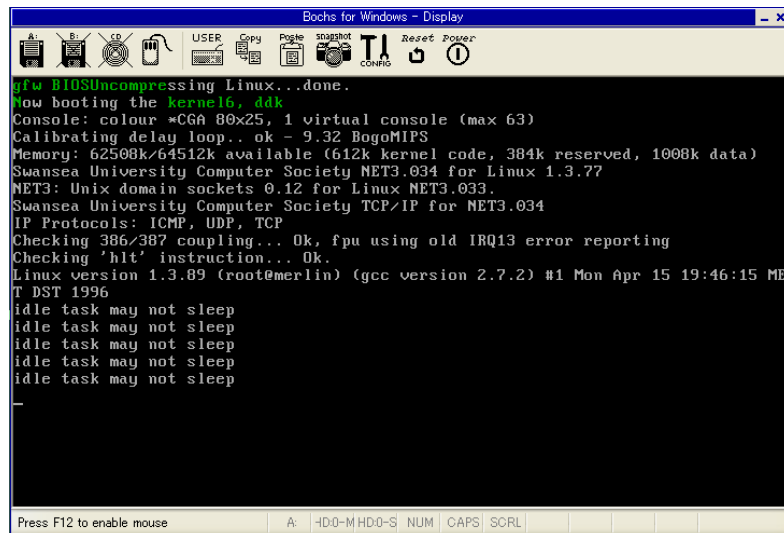


図3.ゲストOS(DLXLinux)のスクリーンショット

また、当初の目的であった「リモートデバイス」を実現するため、Ethernetによる送受信も実装されており、VMはIEEE 802.3 Rawフォーマットのフレームも送受信できる。対応NICはNE2000互換のNICをサポートしている。VMはゲストOSから発行されたin, outといったI/Oを操作するCPU命令を監視することができる。そして特定のI/O命令に関しては、VMがその命令を横取りして、Ethernet上に存在する任意のホストに発行することも可能である。もちろん通常のI/O命令のように自ホストに透過させ実行することもできる。図3のDLXLinuxの起動中では、I/O命令は、すべてローカルホストに透過させている。

4. 従来の技術(または機能)との相違

(開発したソフトウェアの新規性、類似のソフトウェアと比較した場合の優位性等を記述)

PC エミュレータは、大まかに分けて二種類に分けられる。一つは「フルPC エミュレータ」で、CPU のすべての機能をソフトウェアでエミュレーションする方法である。CPU が持つサイクルを忠実にソフトウェアに書き起こす方法なので、正確なエミュレーションを行うことができる。しかしながら当然すべての処理をソフトウェアで行うため、速度が遅くなり、実用にならないものが多い。一方「VMware 式エミュレータ」では、一部の特権命令のみをエミュレーションし、それ以外は直接CPU に実行させるという方法である。エミュレータで動作させることのできるOS は、特権命令を含む以外にはなんら通常のプログラムとは変わりはない。適切なメモリのセッティングと、特権命令の処理さえうまくやれば、そのほかすべての命令は直接CPU で実行してもかまわない。当然OS を構成する命令群のほとんどすべては非特権命令であるので、大幅なパフォーマンスアップが期待できる。しかしながらフルPC エミュレータとは異なり、一部で正確なエミュレーションが行えないといった欠点もある。現時点では完全なエミュレーションではなく、バグも残っているが、ゲストOS に一切手を加えることなく、ユーザモードにて起動できる点は、大きな成果だと思っている。最後に商用PC エミュレータ最大手の「VMware(式エミュレータ)」との相違点について記す。本開発成果と「VMware」での特徴的な違いは次の2 点である。

1. PC/AT 互換機上直接実装されている。
2. CPU の in, out 命令を Ethernet で接続された他の端末に送信可能。

特に 2. の部分については商用製品でも実現されておらず、本開発成果の特徴である。

5. 期待される効果

VMはゲストソフトウェアとしてOSを扱うので、本VMが常駐している限り、ネットワークを介して接続された計算機デバイスを、ローカル側の計算機から自由自在に操ることが可能になる。また、IntelのVTテクノロジーを利用することにより、副次的な効果として一台のPC上で複数のOSを動作させることも可能になる。

6. 普及(または活用)の見通し

本ソフトウェアは実験的なソフトウェアといった色合いが強く、まだまだ実用レベルのものではない。未踏ユースの期間を利用して得られた土台を踏み台としてさらにステップアップを目指すつもりである。私は今後も本ソフトウェアの開発を続けていくつもりである。

今後はIntelのVTテクノロジーを導入して、さらに正確なエミュレーションと機能アップを果たす予定である。

7. 開発者名(所属)

高橋一志 金沢工業大学 情報工学科 2 年

(参考)開発者 URL

<http://d.hatena.ne.jp/ddk50/>

[記載要領]: 開発者の方々に問題のない範囲で下記要領にしたがって記載下さい。

書式: A4(縦置き)で3~4枚程度、フォント(12ポイント、ゴシック)、

余白(上下左右とも25mm)、文字数等(37文字、38行)、

ページ数(1/1、1/2...とし、センタリング設定)。

ただし、テーマ名及びサブタイトルのみ14ポイント、センタリング設定。

テーマ名: 契約名を記載。

サブタイトル: 20字以内で記載(成果のイメージをつかんでもらうためのもの)。

1. ~ 6. までは提案書等をベースに記載。

7. 開発者名: 開発者名に加え、()書きで問合せ先となる所属等を記載。

なお、開発パートナー(会社)についても全て記載。

成果物を紹介する、または成果物に関連するURLがある場合には、URLを記載。

図表: 原則、図表は本文で引用した箇所に近い場所に添付(原稿末尾にまとめて添付することは避けてください)。

図表の前後には、空白行を1行設け、「表」のキャプションは表の上に、「図」のキャプシ

ヨンは図の下に記載。図番号・表番号はそれぞれの通し番号として記載。

ファイル名:「契約番号(詳細)」をファイル名として保存。ex:2004 情財第 100 号(詳細)

* 参考:平成 15 年度成果報告集 <http://www.ipa.go.jp/about/jigyoseika/index.html>