

■仮想化 (Virtualization)

●インターネットと仮想

インターネットコミュニティ上の「仮想世界」、仮想世界で本物のお金と同じように機能する「仮想通貨」、仮想世界での自分の分身である「アバター」などが「仮想」に関連した言葉として一般的にも浸透しています。

「クラウドコンピューティング」という言葉も流行っています。こちらにもサーバ、ストレージ、ネットワークなどのハードウェア資源をインターネット環境で仮想的に扱うための概念です。

仮想化の特徴は以下になります。

- (1) 仮想化する前の実体（リソース）が既知である
- (2) 仮想化された後の実体は仮想化される前の実体と同じように見える

●仮想化の分類

このようにコンピュータの世界での「仮想」とは物理的な何かをシステム上置き換えた「代替物」であるということが出来ます。そして、「仮想化」は「仮想」を扱うための仕組みです。

ここでは「仮想化」を以下のように分類します。

- (a) ユーザインターフェースの仮想化
- (b) 言語処理系の実装表現としての仮想化
- (c) ハードウェアリソースの仮想化

インターネットコミュニティ上の「仮想化」はユーザインターフェースの可視化のためにアバターを定義して、その仮想世界の提供者の収益を上げるために仮想通貨を定義しているに過ぎません。

言語処理系としてJava言語に対するJavaVM (Java Virtual Machine) は「仮想」という言葉が使われています。しかし、JavaVMは実装的視点では単なるインタープリタやコンパイラによるコード翻訳の一つの形態と見なすことが出来ます。Javaバイトコードは独自に定義されたコードであり、他の既存のCPUの命令コードを置き換えたもの（エミュレートしたもの）ではありません。

●仮想化の実装方式

ハードウェアリソースの仮想化の例として「ハイパーバイザ (hypervisor)」という技術があります。この技術ではJavaVMと同じように「仮想マシン (Virtual Machine)」という言葉が使われます。

ハイパーバイザはCPUを含むハードウェアリソース一式を仮想化して仮想マシンを定義します。仮想マシンは仮想マシン上で動作するソフトウェアから見るとハードウェア実体そのものに見えます。仮想マシンは独立していてそれぞれが別々のOSを動作させることが出来ます。この時、実体としてのハードウェアは一つで構いません。ハイパーバイザはハードウェアと仮想マシンの間に介在してエミュレーション動作することになります。

「仮想化」には以下の種類があることになります。

- (A) 可視化の手法としての仮想化
- (B) 変換・翻訳機能 (トランスレータ)
- (C) 既知の何かに見せかける機能 (エミュレータ)

●組み込みシステムと仮想化技術

組込みシステムの開発を中心に考えた時、有効となる仮想化技術はトランスレータ (translator) とエミュレータ (emulator) です。

仮想化階層を利用する側から見ると従来から実在するハードウェアやソフトウェアに対するインターフェースと同等に機能することが必要です。実際には仮想化階層の先にある物理インターフェースとの機能差の程度によってトランスレータとエミュレータは使い分けられることとなります。

単純な手順の変換だけで済む場合はトランスレータで完結することになりますが、複雑な機能の場合はエミュレータとして実装することとなります。

ターゲットとなる実機のソフトウェア構造にこのような仮想化の仕組みが組込まれていればハードウェアリソースに対するアクセス処理は共通に記述することが出来るためハードウェアの物理的な属性や拡張に依存しません。ハードウェアの物理的属性に依存する部分は仮想化技術階層部分で吸収することができます。結果として開発効率、再利用性が上がり、品質向上にも寄与します。また、実機をエミュレートする仮想的な実機がPC上で動作すれば実機なしに開発およびデバッグを行うことも可能です。

一方で仮想化階層が介在することによるオーバーヘッドを考慮する必要があります。