

7. JavaOSとJava実行環境

Javaの実行環境

Javaで作成されたオブジェクトプログラムは、Java仮想機械のコードの形をとる。これをJava byte codeと呼ぶ。Java byte codeの実行は、図-1 (1) にあるように任意のOS上でインタプリタを作動させる方式（たとえば多くのウェブブラウザの利用の場合）が基本的なものであった。これは性能的にはあまり高いものではない。そこで、いくつかの性能向上の手法が開発されている。

Javaのそもそもの発端であるセットトップボックスあるいは情報家電といった実行環境を考える場合には、汎用のOSは必ずしも必要がなく、図-1 (2) のようにJavaコードの実行を目的としたJavaOSを置くことが提案されている。また、Javaの実行だけを目的とするネットワークコンピュータにおいても同じ形態が志向できる。

汎用のコンピュータ上の汎用のOSのもとでの実行を高速化させるには、byte codeに対してコンパイラを実行させて、その機械の機械語に変換してから実行する図-1 (3) に示す方式が有効と考えられている。実行の直前にこれを行なうJust In Timeコンパイラと、前もって実行させておくスタティックコンパイラの2つにさらに分かれる。

Java byte codeの実行に対する究極的な方式は、byte codeを直接実行するコンピュータを用意することである。図-1 (4) がそれであり、いわゆるJavaChipを用意し、その上にJavaOSを搭載する。もし、その環境におけるすべてのアプリケーションがJava環境ですむのであれば、この方式は最適な選択となり得る。一方、汎用性/拡張性などの点で、今後の推移を見守る必要がある。

以下に、JavaOSについて解説する。

JavaOSは何のため？

Java OSは、Javaアプリケーションの実行のためのOSである。他の言語で書かれたアプリケーションについては考えられていない。イントラネット、インターネットおよび組み込み装置を意識して設計されている。また、JavaOSは、現在までのところ、JavaChipのためのOSに限定したものではない、複数のハードウェアプラットフォームをターゲットとしている。JavaOSは、こうした環境の中で、クライアント管理の労力を軽減できるOSとして考えられている。

JavaOSは、小さく効率のよいことが意図されており、4MBのROMと4MBのRAMを最小実行環境として設定している。普通のパソコンが32MBあるいは64MBの主記憶を実装する時代に、この小ささの強調は、「制限されたリソース環境」を意識したものである。

また、JavaOSは、いわゆるネットワークコンピュータのOSとして注目を浴びた。いったんネットワークに接続されれば、さまざまな機能を発揮するように意図されている。さらにその場合、HotJavaブラウザによるウェブへの対応、HotJava ViewsによるGUI環境がベースとして用意される。

JavaOSの概要は、<http://java.sun>。

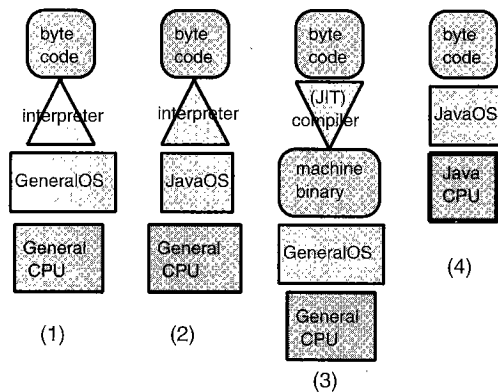


図-1 Javaの実行環境4態

com/marketing/collateral/os.htmlおよび
http://java.sun.com/products/javaos/index.
htmlなどから知ることができる。

JavaOSを搭載する機器は低価格で管理のいらな
いステーション (Zero Administration Client) と
いうのが特徴といわれてきた。しかし、普通のデスク
トップを必要とする環境では、それらが今までにもつ
ていたさまざまなソフトウェア環境の移植あるいは互換
性の保証が必要となる。ハードウェアリソースへの対
応も重要になる。このため、JavaOSを搭載したコン
ピュータは汎用のコンピュータというよりも、専用
の機器としての場合の方が、より有効性を発揮でき
るものと考えられる。

このような流れは自然に、組込み機器用のOSとい
う色彩を強く持つものへと向かうことになる。いわゆ
る情報家電への期待である (図-2)。

JavaOSのアーキテクチャ

JavaOSはレイヤードアーキテクチャによってい
る。大きくはプラットフォーム依存部とJavaで書かれた
プラットフォーム独立部の2層からなる。

(1) プラットフォーム依存部：プラットフォーム依
存部は、その機械の機械語でコードされる。マイク
ロカーネルとJVMがその中にある。

マイクロカーネルには、ブート機能、割り込み処理、
多重スレッド機能、トラップ、DMA機能がある。
Sun Microsystemsからのライセンスによりリリース
されている現在の版では、SPARC, x86,
StrongARMがサポートされている。なお、これらの
開発には、GCC 2.7.2をベースにした開発環境が想定
されている。JVMは、Java bytecodeのインタプリタ
ループと実行だが、このためにメモリ管理、スレッド
の制御、クラスローディング、およびbytecode veri
fierがその中に含まれている。

(2) プラットフォーム独立部：プラットフォーム独
立部には、以下のような機能が入れている。

JavaOS Window system, JavaOS Graphics sys
tem, JavaOS device drivers, JavaOS Network
Classes

JavaOS Window systemは、ボタン、メニュー、

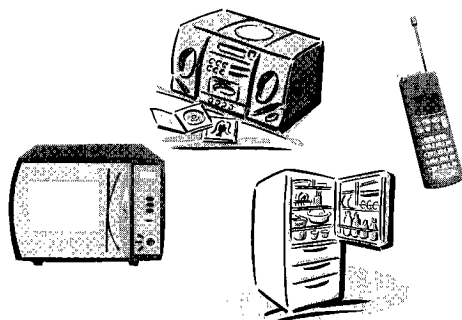


図-2 情報家電への期待

スクロールバーなどGUIコンポーネントを供給する。
また、オーバーラップウィンドウに対応している。

JavaOS Graphics systemは描画パッケージを供給
する。

JavaOS device drivers (デバイスドライバ) は
Javaで書かれ、移植性があり、拡張性がある。

Network Classesは、TCP/IPをサポートし、ネッ
トワークトランスポートおよびルーティング機能を供
給する。DNSおよびNIS機能があり、ホスト名参照、
ユーザなどに共通した機構を用いることができる。
RARP, DHCP, NFS client, SNMPが組み込まれ
ている。SSLサポート、PPP, HCP起動時のベンダ
オプションデモ用のセットアップなどがあり、ネッ
トワークに関連する機能がいろいろと具備されている。
その詳細は省略する。

JavaOSは、マルチプログラミングではないが、マ
ルチスレッドに対応している。つまり、アプリケーション
実行中にネットワークへの対応をしたりすることは
できる。

JavaOSの起動

JavaOSは、ROM、ネットワーク、そして、ハー
ドディスクなどから起動できる。ネットワークコンピ
ュータとしての利用の場合には、おそらくはネット上
のサーバからシステムをダウンロードして起動する
ことが考慮されるだろうし、組込み機器・ハンドヘルド
の場合にはROMから立ち上がることが主に考慮され
るだろう。

JavaOSでのJava言語仕様

JavaOSは、Javaの言語仕様の水準を特定しない。
JavaOSで動作するアプリケーションは、Javaが動作
できるブラウザ・OSでも動作する。けれども、Java
Application Environment (JAE) という概念があ
る。たとえば、対象とする実行環境がEmbedded
JavaあるいはPersonal Javaの範囲だとすると、グラ
フィック関連のサポートがなくなり、それらを要求す
るプログラムは動かなくなってしまう。どの範囲のク
ラスライブラリが必要なのかによって、実行環境の水
準分けをする仕組みの必要性が出てくる。これがJAE
である。

Embedded Java/Personal Javaという仕様水準の
設定は、JavaOSの開発開始のあとから出てきた概念
である。つまり、小さな応用には小さな機能しかいら
ないということを明確化できる領域があるということ
であり、歴史的な理解としてうなづけるものがある。
しかし、仕様と対象領域の対応がはっきりとしたもの
となるかは、今後の実用化により答えが出る。

(平成10年3月9日受付)