

第4章

組み込みシステムの ためのJava実行環境と リアルタイムへの対応

— JTRONと開発環境

根本 忍

組み込みシステムではリアルタイム性が要求されることが多い。しかしJavaにはリアルタイム処理のための機能がないので、現在の仕様のままでは組み込みシステムへの応用が難しい。この問題を解決するため、JavaをサポートしたリアルタイムOSがいくつか提供されている。この章では、リアルタイムOSの一つであるITRONとJavaを融合したJTRON仕様について紹介する。とくに、Javaにリアルタイム性をもたせるしくみを中心にとりあげる。(編集部)



はじめに

家電製品/OA機器から携帯端末、カーナビ、工業用制御機器にまで広がる組み込み分野で使われているソフトウェアは、いわゆる組み込み用リアルタイム・オペレーティング・システム(以後、リアルタイムOS)や各社で独自に開発されたOSがほとんどである。互換性がなく、ネットワークを利用した協調動作やグラフィカル・ユーザ・インターフェース(GUI)などの機能ももっていない。

最近になって、Javaが組み込み用として注目されはじめている。もともとJavaは、「Green Project」において組み込み制御用言語「oak」として設計された経緯があり、これは自然な成り行きともいえる。

Javaには、ポータビリティやGUI、ネットワーク機能といった大きなメリットがある。反面、厳しい制限とさまざまな要求にさらされる組み込み用として用いるには、いくつかの問題もある。たとえば、ガーベジ・コレクションのために犠牲になったリアルタイム処理機能、ハードウェアの制御、そしてバイト・コー

ド・インタプリタで動いていることによる動作の遅さなどがあげられる。いずれも、現在のJava単独で解決できるものではない。

そこで本稿では、組み込みシステムにおけるJavaの現状と現実的なソリューションとしてのJTRON、およびその実装について解説する。

組み込みシステムとJava

Javaには、プログラム言語および実行環境としての二つの側面があるが、いずれも、単独で組み込み用途に用いることは非常に困難である。

言語レベルから見れば、ポインタを持っていないのでデバイス・ドライバや割り込みハンドラの記述ができない。実行環境としてみても、マーク・アンド・スイープ方式によるガーベジ・コレクションを採用しているため、リアルタイム・システムにとっては致命的なスリープ・タイムが発生してしまう¹⁾。

組み込みに使う場合の課題については、さまざまな手法が試みられている。しかし、いたずらにJavaを独自拡張し

たり、JNI(Java Native Interface)を使用してデバイス・ドライバにアクセスしたりすると、プラットフォームからの独立性を損なうおそれがある。Javaの特長を活かして分散処理するようなアプリケーションおよびアーキテクチャのデザインが必要になる。

○Javaの実行環境(1)

～ Sun Microsystems 社

Sun Microsystems 社(以後、Sun 社)のJavaの実行環境には、公表されているものだけで以下のようなものがある(図1)。

(1) Java(JDK)

(2) JavaOS

JavaOS for Business

JavaOS for Consumers

JavaOS for NCs

(3) PersonalJava

(4) EmbeddedJava

(5) JavaCard

(6) JavaPC

(7) Java Virtual Machine(JVM)

ただし、これらのなかには、単独では製品として提供されていないものも含まれている。

実際に組み込み用途への導入が進みつつあるものとしてはJavaOSとPersonalJava(図2)があげられる。しかしながら、機能が拡充するにつれ肥大化する傾向に



【図1】Java技術に関する情報 - Sun Microsystems 社のホームページ <http://www.java.sun.com/>



【図2】PersonalJavaに関する情報 - Sun Microsystems社のホームページhttp://www.java.sun.com/products/personaljava/

あるばかりでなく、組み込み用途で求められるリアルタイム処理やハードウェアの制御機能を備えていない。

PersonalJavaにしてもEmbeddedJavaにしても、フルJava(JDK)から組み込み用としては必要ないと思われる機能を抜いたり、グラフィックス機能を落としてフットプリントを小さくしているだけで、時間的制約を実装するためのしくみはもっていない。したがって、そのまま機器に組み込むには無理がある。

本来組み込み用としてアナウンスされていたのはEmbeddedJavaであった。しかし、リリースの遅れなどから、サード・パーティがEmbeddedJavaをターゲットにしたJavaクローンの開発を進める結果となった。後で詳しく説明する「ChaiVM」がその代表的なものである。

○リアルタイム化への動き

すでに触れたように、予測不能なガーベジ・コレクションがあるため、根本的にリアルタイム・システムでJavaを使うことはできない。

最近では、Sun社自身による1998年11月3日のアナウンス「Real-Time Extensions for Java Platform」²⁾も含め、いくつかJavaのリアルタイム化についての動きがあるが、いずれも製品として提供される時期などは明らかにされていない。



【図3】ChaiVMに関する情報 - Hewlett Packard社のホームページhttp://www.chai.hp.com/emso/

い。また、1998年12月8日にJava Business Expo(JBE)で発表された新ライセンスSCSL(Sun Community Source License)に基づいてソース・コードが公開される「Java2(JDK1.2)」についても、ビジネスとしての実情がまだ見えていない状況である。

○Javaの実行環境(2)

~ Hewlett-Packard社

Hewlett-Packard社(以後、HP社)は、組み込み用途向けのJavaソリューションの提供では、Sun社を一歩リード

している。1998年3月に「HP Embedded Virtual Machine」として発表され、のちに「ChaiVM」と名称が変更された独自開発によるJava実行環境は、すでにMicrosoft社をはじめ多くのリアルタイムOSベンダに採用されている(図3)。

ChaiVMは、完全に独自開発された組み込み用途向け(small-footprint implementation)のJavaVM準拠の実行環境である。ソース・コードは約7万行で、C++で実装されている。

1998年11月2日には「ChaiVM Release 2.0」が発表された。Release 2.0には、セキュリティ機能の強化をはじめとして、表1のような特徴がある。このほか、組み込みに適したグラフィックスとして「HP ChaiAWT」がオプションで提供されている。ChaiAWTパッケージは、java.awt.peerパッケージとフル機能のGUI両方の完全な実装を含んでおり、リニア・メモリ領域に直接描画できるようになっている。

ChaiVMのアーキテクチャを図4に示す。

○EmbeddedJavaとChaiVMの違い

HP社のChaiVMがSun社のEmbeddedJavaと最も違うところは、開発者が実行環境として用意すべきクラス・ライブラリの構成を自由に変更できるライセンス形態とライセンス料金にしてい

【表1】ChaiVMの仕様

ChaiVM Release 2.0	完全バイト・コード・インタプリタ
	JNI(Java Native Interface) Exception handling Multithreading capability インクリメンタル・ガーベジ・コレクション JDK 1.0.2(with 1.1 signatures) java.lang, java.io, java.util, java.net クラス・ライブラリ JDK 1.2セキュリティ・マネージャ(permission-based subset; optional) ダイナミック・クラス・ロード/アンロード(オプション) ダイナミック・ネイティブ・メソッド・ローディング(オプション) Small footprint 680 KバイトROM(StrongARM用にすべての機能を実装した場合) 最大265 KバイトRAM
ツール/ユーティリティ	コマンド・ライン・リモート・デバッガ クラス・ファイル・チェッカ JNIヘッダ・ファイル・ジェネレータ
動作環境	ターゲットCPU StrongARM, MIPS-4300(他CPUへの移植も可能)
	開発環境 HP-UX/PA-RISC, Windows NT(x86)