

T-Engine の思想

坂村 健

組み込み機器の世界では従来、ITRON が広く使われてきた。コンパクトな仕様で知られ、多数の実装をもつ ITRON は、数多くの機器に組み込まれ、社会を支えている。

ここに新たに T-Engine というプラットフォームが投入されようとしている。ハードウェアの仕様を固め、ミドルウェアの流通を容易に行おうとする実践的な試みだ。

そこで本特集の第 1 章として、TRON プロジェクト・リーダーの坂村 健氏が、T-Engine のめざすところについて解説する。

(編集部)

はじめに

TRON プロジェクトは 1984 年にスタートし、今年でちょうど 20 周年を迎えた。TRON プロジェクトが最初に手がけた組み込み機器向けのリアルタイム・オペレーティング・システム (リアルタイム OS) である ITRON は、今や、携帯電話をはじめとして、情報機器や家電、車のエンジン制御などに広く利用されている。

さて、本特集で取り上げる T-Engine と T-Kernel であるが、今までの ITRON との違いは何かという質問をよく受ける。本特集をもって、T-Engine の考え方を理解していただければと思う。

ご承知のように携帯電話は爆発的に普及し、モデル・チェンジが頻繁に行われ、そのたびに新しい機能やアプリケーションが追加されている。組み込み機器にもかかわらず、そのソフトウェアのステップ数は 100 万ステップを越えるという大規模なものだ。開発サイドから見ると非常に短期間で高度なソフトウェアを完成させなければならない。さらに、“ユビキタス・コンピューティング”という新しい応用が生まれつつあり、ソ

フトウェア開発への要求は減ることはなく増えつづけている。ソフトウェアの生産性をいかに上げるかというのが組み込みシステムに対する最大のテーマである。

ソフトウェアの生産性を上げるためにはいくつかの方法があるが、一度作ったソフトウェアを再利用していくというのが一番わかりやすい。しかし、現実にはソフトウェアの再利用というのはなかなか難しく、機器内部のハードウェア構成が変わり、モデルが変わるたびに作り直しということが多い。

この問題を解決し、一度作ったソフトウェアは 100 年間利用しようということから「100 年ソフト」というキャッチ・フレーズを掲げて研究開発を進めているのが T-Engine 開発プラットフォームであり、リアルタイム OS の T-Kernel である。



ITRON からの教訓

ITRON を開発したころは、まだマイクロプロセッサが出始めの時期であった。ハードウェア・リソースの規模も小さく、OS はほとんど利用されていない状況だった。ソフトウェアは OS なしで、アプリケーションが I/O から割り込みまでのめんどろを見るように毎回作り込まれていた。

よく使われる処理や割り込み処理、すなわち汎用リアルタイム OS の部分を提供しようというのが ITRON の発想であるが、マイクロプロセッサの能力が低かったため、OS の標準化に際して「弱い標準化」という考えを導入した。32 ビット・マイクロプロセッサの領域までを考慮して豊富な機能を用意しつつも、どこまでの機能を用意するかをクラス分けし、マイクロプロセッサのリソースや規模により、クラスを選べるようにした。また、マイクロプロセッサの事情により実装依存を許す部分を設けることにより、多様なマイクロプロセッサに対応できるようにした。その結果、ほとんどのマイクロプロセッサに ITRON が実装され、多くの開発者が ITRON を利用し、組み込み機器向けリアルタイム OS としてデファクト・スタンダー

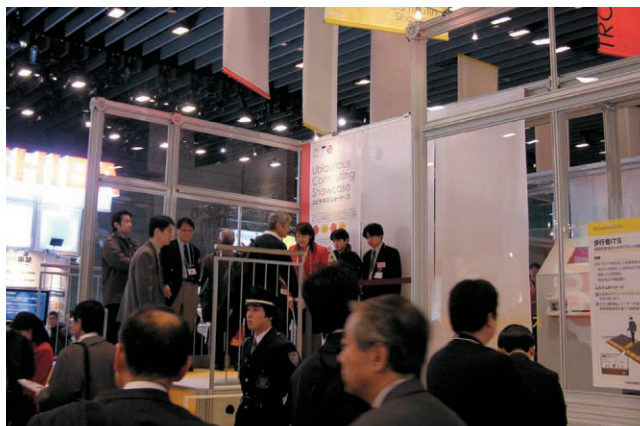


写真 1 TRONSHOW 2004 のようす

ドになった。

ところが「弱い標準」という考え方は、A というマイクロプロセッサ用の ITRON と、B というマイクロプロセッサ用の ITRON は似ているが違う部分もあるということである。PC の世界の Linux にも多くの似て非なるものがあり、その違いによってアプリケーションがうまく動かないということがあ。同様に、ITRON の実装による微妙な差は ITRON の上で動くミドルウェアやアプリケーションの作り方に影響し、ほかの ITRON で動かなかったり、移植するためにはむだなコストや時間を負わなければならないということにもなった。



T-Engine の発想

T-Engine の発想は「ミドルウェア流通のプラットフォーム」である。このためにリアルタイム OS をシングル・ワン・ソース (ただ一つしかないソース・コード) として提供し、その上で動作するミドルウェアが流通しやすいようにした。もちろん、CPU が異なればバイナリのままでミドルウェアが動くわけではないが、再コンパイルにより利用できるようにしようという考え方である。このシングル・ワン・ソースの OS が T-Kernel である。そして、T-Kernel を動かすための標準ハードウェア・プラットフォームが T-Engine である。

T-Kernel はミドルウェアの流通を考え、従来の ITRON が持つスタティック・メモリ・アロケーションに、ダイナミック・メモリ・アロケーションを加え、タクスやセマフォなどのリソース ID を自動で割り当てできるようにしたものだ。さらにファイルやプロセス管理をサポートする上位層として T-Kernel Extension を用意した。現在、T-Kernel 用のミドルウェアの流通を促進するために、T-Engine フォーラムでミドルウェアをデータベースに登録し、eTRON と呼ぶセキュリティ・アーキテクチャを利用して、ミドルウェア・ベンダとミドルウェアを利用するシステム・メーカの間で、電子的にミドルウェアを流通させる T-Dist と呼ぶしくみを整備しつつある。このようなしくみを用意することによって、システム開発者は、自分の欲しいミドルウェアをネットワークで探し、即座に試し、良ければ購入するというようなことができるようになる。



T-Engine と T-Kernel による開発

T-Engine であれば T-Kernel は必ず動作する。しかし T-Kernel を動かすために必ず T-Engine が必要というわけではない。T-Engine と T-Kernel による開発は次のような流れになる。

T-Engine はすでに主要な組み込み用の CPU を搭載したものが何種類も用意されている。システム開発者は、この中から目的に合いそうな CPU を搭載した T-Engine を入手し、T-Kernel や開発環境を整える。この時点でソフトウェア開発がスタートする。そして、ハードウェア・チームは、T-Engine のハード



写真2 TRONSHOW 2004 で展示された T-Engine

ウェアにない機能や、速度が必要とされるためハードウェア化したほうが良いような機能を FPGA などを使って拡張ボードとして作り、T-Engine ボードと組み合わせることで機能的には最終目的に近いシステムのハードウェア・プラットフォームを完成させる。そしてこれをソフトウェア・チームに引き渡す。

ソフトウェア・チームは、拡張ボードのついた T-Engine を使い、利用できそうな流通ミドルウェアも組み合わせ、ソフトウェアの開発を進行させる。その間にハードウェア・チームはハードウェアの一部を ASIC 化したり、ターゲット向けの基板を作り上げたりする。そして、並行して開発しているソフトウェアを搭載して最終製品にしていく。

T-Engine を利用することにより、最初の開発ハードウェア・プラットフォームをつくりあげるまでの時間とコストを下げることができる。また、ソフトウェアの開発途中でも、T-Engine を利用してデモンストレーションを行うことができ、マーケティングなどの意見をフィードバックするのに大いに役立つ。従来の組み込み機器用の開発ボードは大型のものが多かった。何度も繰り返すが、T-Engine のハードウェアはあくまで開発のためのプラットフォームである。しかし、コンパクトに作ることにこだわったため、開発途中でも製品イメージにできるだけ近いものが作れるようになった。



T-License

T-Kernel はすでに T-Engine フォーラム (<http://www.t-engine.org/>) よりソースを配布している。T-Kernel のソースの配布を受けるには T-License と呼ぶ契約を締結してもらえば、その条件でだれでも (T-Engine フォーラム会員でなくても) 無償で組み込み機器の製品に利用することができる。

ソースをオープンにしているソフトウェアはいくつかあるが、それぞれ利用するための条件が異なる。たとえば GPL では、バイナリを配布した人に対して、改変したソース自体を公

T-License

T-Kernel の配布に際しては T-Engine フォーラムにより定められた T-License が適用される。このライセンスを以下に挙げる。

T-Kernel ライセンス

T-License [T-Kernel のソースコードのライセンス契約]

T-Engine フォーラム

第1条 規定範囲

1. 本ライセンス契約は、T-Engine フォーラムより配布を行う T-Kernel のソースコード及びその派生物に関する、著作権ならびに利用条件を定める。

第2条 用語定義

1. 「T-Kernel」とは、著作権者より T-Engine フォーラムが委託を受け管理および配布を行う T-Engine 用リアルタイムオペレーティングシステムをいう。
2. 「本ソースコード」とは、T-Kernel のソースプログラム（付随するコメント、ドキュメンテーションを含む）をいう。
3. 「単純移植されたソースコード」とは、T-Engine フォーラムに登録された T-Engine ハードウェアで稼働するように、T-Engine フォーラム発行の T-Engine ハードウェア仕様書に準拠して製作され、本ソースコードのハードウェア依存部のみを改変したものをいう。このうち、T-Engine フォーラムに登録されたものは「本ソースコード」に含める。
4. 「改変されたソースコード」とは本ソースコードを性能強化、機能追加・削減などを目的として改変して生成されたソースコードをいう。「単純移植されたソースコード」は「改変されたソースコード」の定義には含まれない。
5. 「バイナリコード」とは本ソースコードまたは改変されたソースコードの全部もしくは一部を含むプログラムをコンパイルして生成された実行形式のコードをいう。
6. 「本ソースコードの派生物」とは、改変されたソースコードとバ

イナリコードを総称したものをいう。

7. 「組み込み製品」とは、本ソースコード、改変されたソースコード、またはバイナリコードを利用し、ハードウェアに実行形式のコードを搭載して作動する機器をいう。
8. 「最終利用者」とは組み込み製品を購入して使う一般の消費者をいう。
9. 「システム開発者」とは、組み込み製品を自らまたは第三者に委託して開発し、最終利用者に組み込み製品を有償、無償を問わず提供する者をいう。
10. 「改変版配布者」とは、改変されたソースコードを製作し、有償無償を問わず第三者に配布する者をいう。
11. 「改変版パッチ」とは、本ソースコードから改変されたソースコードあるいはそのバイナリコードを生成するための差分あるいはその生成プログラム、生成システム等をいう。
12. 「パッチ処理の代行」とは、本ソースコードに対して改変版パッチを利用して改変されたソースコードあるいはそのバイナリコードを生成する作業を代行することをいう。
13. 「ソースコード利用者」とは本ソースコードを利用する者をいう。
14. 「配布」とは、次のことをいう。

1. インターネット等の通信、放送等により、著作物を特定多数の人に送信すること。
2. インターネット等の通信、放送等により、著作物の送信を不特定の人からの求めに応じ自動的に行うこと。
3. 著作物の複製物を、不特定または特定多数の人に頒布すること。

第3条 本ソースコードの著作権

1. 本ソースコードの著作権は坂村健が有する。

第4条 利用許諾

1. T-Engine フォーラムは、T-Engine フォーラムの定める所定の登録手続を済ませ、かつ T-License に同意した者に対して、以下に定める通り、本ソースコードを無償で利用許諾し、提供する。
2. 本ソースコードは T-Engine フォーラムのみより配布を行う。本ソースコードを入手した者は本ソースコードを再配布してはならない。

開しなけなければならないというような条件がある。組み込みシステムを考えた場合、改変した部分というのはノウハウに属して公開したくないということが一般的である。T-Kernel の配布のためのライセンスは、組み込み機器での利用に適するようにした T-License と呼ぶ契約となっている（コラムを参照）。

T-License では、ソースを自由に改変してそれを機器にバイナリの形で組み込んで製品として販売することができ、そのためのライセンス費用も無償である。

おわりに

T-Engine はユビキタス・コンピューティングの世紀を目前と

して組み込み機器の開発をいかに効率よくするかをテーマに開発、標準化を行ってきた。当初 22 社の賛同で始まった研究開発の中心的組織である T-Engine フォーラムも、すでに海外国内あわせて 410 社という会員の参加を得ている。シングル・ワン・ソース、ミドルウェアの電子流通メカニズム、組み込みシステムに適した T-License など、次世代組み込み機器の生産性向上に T-Engine が大きな貢献をできることを信じている。

さかむら・けん 東京大学教授、TRON プロジェクト・リーダー