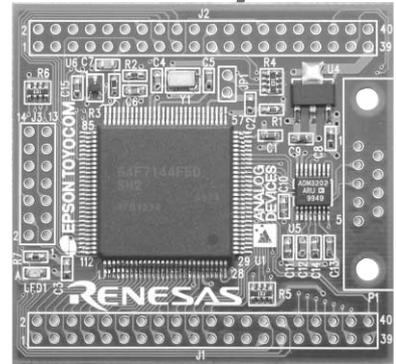


## μITRON4.0仕様のツボを押さえる TOPPERS/JSPを 理解するための μITRON4.0仕様



邑中 雅樹

μITRONはUNIXやWindowsなどのOSと比較すると、数々の特徴をもつ。それは組み込みでの使い勝手を優先した結果、省資源やリアルタイム性といった目的を達成するために導入された概念だ。

本章では、TOPPERS/JSP、ひいてはμITRON4.0仕様を理解するための背景から解説を行う。(編集部)

本章では、TOPPERS/JSP カーネルのベースとなっている μITRON4.0仕様スタンダード・プロファイルの全体像について説明します。

μITRON4.0仕様は、その仕様書の中に【仕様決定の理由】という項目があり、ほかのオペレーティング・システム仕様よりも「なぜそうなっているのか?」ということがわかりやすい構成になってはいます。しかしそれも徹底しているわけではなく、当時カーネル仕様の検討WG(Working Group)のメンバだった方々に直接聞いて初めて理解できることも多々あります。

本章では、μITRON4.0仕様書を読み解く前に知っておいたほうが便利そうな話題をなるべく多く取り上げました。なお、μITRON4.0仕様の最新版であるμITRON4.02仕様書は、無償でダウンロードできます<sup>注1</sup>。興味のある方は、この機会にぜひご一読ください。

### オペレーティング・システムの3大機能

少し大局的にオペレーティング・システム(OS)というものを考えてみましょう。OSが提供する機能とは何でしょうか。一般論として、ハードウェア・リソースの抽象化、抽象リソースの管理、リソースの利用効率の向上、の三つがあると言われています。

上記の三つとは独立ではありませんが、いまだきの存在理由として、抽象リソースのアクセス保護、を強調する意見もあります。

世の中には、目的別にたくさんのOSが存在しています。そのいずれも、目的別に最適化された形で、上記の～の三つの機能を提供しています。

ここで、～の機能いずれも「リソース(資源)」ということばが関与していることに注意してください。つまるところ、OS

の機能とは、有限個しかないリソースを複数の仕事を取り合いにならないよう、最適に割り振るための調停役として存在します。

### μITRON4.0仕様における方針

続いて、これらの機能について、μITRON4.0仕様はどのような方針をとっているのか、順に見ていきましょう。

#### ●ハードウェア・リソースの抽象化

LinuxやWindowsなどデスクトップ由来のOS、およびQNXやVxWorksなどPOSIXの影響を受けている組み込みOSは、周辺デバイスも含めたシステム全体のハードウェア抽象化機能を提供しています。そのような世界では、デバイス・ドライバを定義するモデルが存在し、アプリケーションからは統一された方法でたとえばopenシステム・コールを使ってファイルをオープンするのと同じ方法で、抽象化されたハードウェア・リソースにアクセスすることができます。

しかし、μITRON4.0仕様が提供するハードウェアの抽象化とは、CPUコアに限定した、緩い抽象化にとどまっています。

TOPPERS/JSPカーネルの開発者であり、μITRON4.0仕様の編者でもある名古屋大学の高田広章氏は、「ITRONとはCPUコアのデバイス・ドライバである」という表現をすることがあります。これは、言い換えるとCPUコア以外の周辺デバイスはμITRON仕様OSの管理の外であるということを暗示しています。

μITRON仕様は、4ビットCPUから64ビットCPUまで、幅広い範囲で使われる可能性があります。接続される周辺デバイスもさまざまです。下手にハードウェアを抽象化すると、採用できないアーキテクチャが出てきてしまうことは想像に難しくありません。そのため、このような緩い抽象化となって考えることができます。

割り込みやCPU例外といった、多くのCPUで存在を仮定できるものも、μITRON4.0仕様ではあいまいな箇所が多数あり

注1: <http://www.ert1.jp/ITRON/SPEC/mitron4-j.html>

ます。CPU 例外はアーキテクチャごとの違いがあまりにも大きく、今後もまとめることは難しいと筆者は考えています。しかし、割り込みに関しては、TOPPERS プロジェクト内で新しいモデルを採用する動きもあります(コラム1参照)。

ちなみに、μITRON 仕様としてはデバイスについてデバイス・ドライバ・ガイドラインのような外部仕様を用意して、カーネルとは別の領域の問題と捉えようとしているようです。

TOPPERS/JSP を始めとする TOPPERS 系 ITRON 仕様 OS では、デバイス・ドライバ・ガイドラインで提案されているシステム・インターフェース・レイヤ(SIL)に基づくデバイス・

アクセスが推奨され、ターゲット依存部も SIL を用いるように書かれています。

### ● 抽象リソースの管理

ここでいう抽象リソースとは、「物理的なものとは直接対応しないがアプリケーションからは操作ができる、状態をもったもの」を指します。抽象リソースはおおむね下記の2種類に分かれます。

- 物理的な複数のリソースを取りまとめ、アプリケーションからは一つのリソースになるように見せかけているもの
- 物理的な存在がないのに、アプリケーションからは存在する

## コラム1 TOPPERS 標準の割り込みモデル

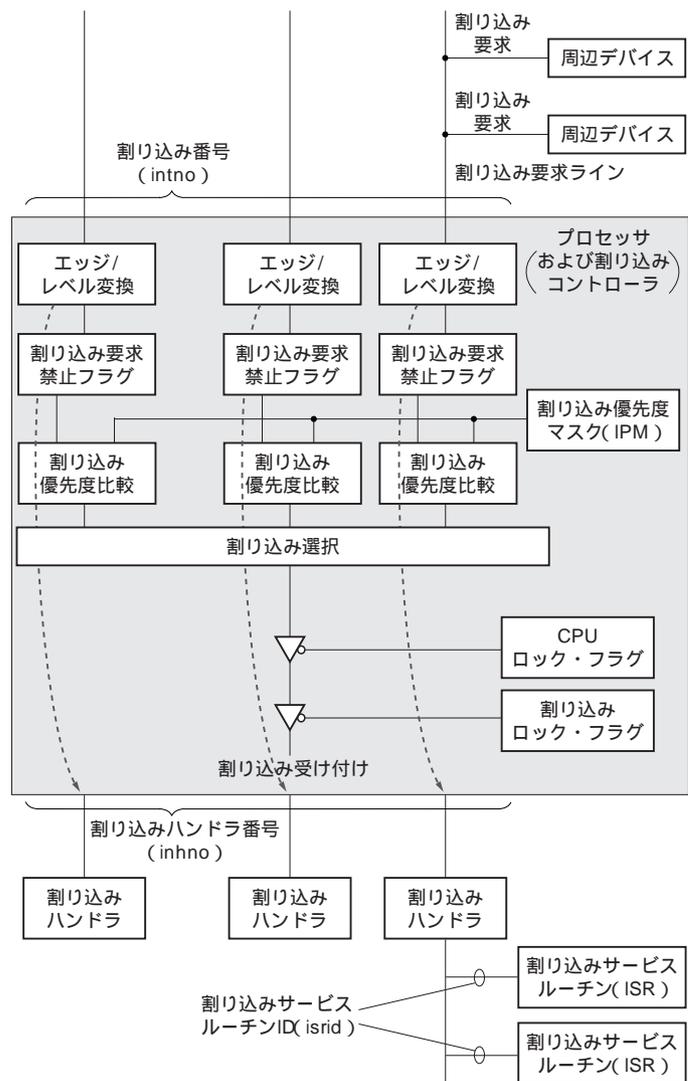
ハードウェアの抽象化レベルを高めない方針は、多くのアーキテクチャで採用できるというメリットがあります。一方で、異なるアーキテクチャ間での移植性が保てないというデメリットも生み出します。移植性が高まらないとソフトウェア部品を作る動機が高まらず、生産性を高めることが難しくなります。ソフトウェア部品については、(社)トロン協会などが主導したデバイス・ドライバ・ガイドラインなどの例が存在します。しかし、デバイス・ドライバを作るときに重要な、割り込み処理に関する抽象モデルは手つかずのままでした。

そこで、TOPPERS プロジェクトでは、TOPPERS/JSP カーネル実装の経験やOSEK/VDX 仕様の分析などを踏まえ、TOPPERS 標準割り込みモデルを作成しています。アイデアの多くが TOPPERS/JSP カーネルで行われている割り込みの仮想化をベースとしているため、仕様の初期案の段階である程度使いものになるものが出てきました。さらに、カーネル開発者を中心とする技術者で会議や合宿を行いながら、なるべく多くのアーキテクチャで性能を落とさずに抽象化できるようブラッシュアップを加えています。

TOPPERS 標準割り込みモデルが想定する仮想ハードウェアのモデルを図Aに示します。概略としては、入力トリガの種類が選べ、優先度付きで、割り込み全禁止ができ、CPUに直結されている、といったそれなりにリッチな割り込みコントローラが仮想的に存在するとみなすこととなります。そのような割り込みコントローラがない場合には、ソフトウェアでエミュレーションすることとなります。バス・ブリッジでつながるバスの外の割り込みコントローラは、管理の対象にはなりません。

いくつかのAPIや概念が追加になり、この割り込みモデルを採用した場合には、μITRON4.0仕様とは完全互換にならない可能性が高くなります。

現在、仕様案はTOPPERSプロジェクト会員向け早期リリースの扱いで、非会員が閲覧することはできません。しかし、近日中にTOPPERS/ASPやTOPPERS/HRPといったカーネル実装と同時に無償公開される予定です。



図A 仮想ハードウェア・モデルの割り込みモデル