

オープンソースの μ ITRON 仕様 OS: TOPPERS/JSP カーネル

(第 3 回 IP アワード 応募書類 特徴の説明書)

高田 広章 若林 隆行 本田 晋也

豊橋技術科学大学 情報工学系

1 μ ITRON 仕様とシステム LSI

μ ITRON 仕様は、(社) トロン協会 ITRON 部会によって策定された組込みシステム用のリアルタイム OS 仕様である。トロン協会の最新の調査結果 [1] によると、最近開発された組込みシステムの 3 分の 1 以上に μ ITRON 仕様に準拠したリアルタイム OS が使用されており、 μ ITRON 仕様はこの分野においてデファクトスタンダードとなっている。とりわけ、通信端末、家電機器、AV 機器、個人用情報機器といったコンシューマ機器分野において μ ITRON 仕様 OS の使用率が高い。このように、 μ ITRON 仕様 OS の適用分野はシステム LSI の主要な適用分野と一致しており、 μ ITRON 仕様 OS はシステム LSI を用いたシステム開発に不可欠な IP (ソフトウェア設計資産) の一つであるといえることができる。

μ ITRON4.0 仕様 [2] は、 μ ITRON 仕様の最新バージョンで、1999 年 6 月に公開された。近年の組込みソフトウェアの大規模化・複雑化により、この分野においてもソフトウェア部品 (ないしは、ミドルウェア) の活用やソフトウェアの再利用が重要な課題となっており、 μ ITRON4.0 仕様は、ソフトウェアのポータビリティを向上させることを最も重要な策定目的としている。具体的には、「スタンダードプロファイル」と呼ばれるプロファイル規定を行い、それに準拠した範囲では、高いソフトウェアポータビリティを実現可能としている。 μ ITRON4.0 仕様に準拠した OS は、すでにいくつかのメーカーによって開発され、製品として販売されているものもある。

2 TOPPERS/JSP カーネルの概要

TOPPERS/JSP カーネルは、 μ ITRON4.0 仕様に準拠したオープンソースのリアルタイム OS である。「JSP」は、「Just Standard Profile」の略称であり、TOPPERS/JSP カーネルは、その名前が示す通り μ ITRON4.0 仕様のスタンダードプロファイル規定「ちょうど」の機能を持つように設計した¹。

¹実際には、スタンダードプロファイル外の機能も若干サポートしている。

ターゲットプロセッサとして、現時点では、モトローラ社の M68040 と日立製作所の SH3 をサポートしており、他にいくつかのプロセッサへのポータリング作業が進行中である。また、Linux および Windows 上で動作させるためのシミュレーション環境を用意している。

TOPPERS/JSP カーネルを開発した一義的な目的は、研究・教育機関における研究・教育への利用と、 μ ITRON4.0 仕様の評価である (すでに、これらの目的で TOPPERS/JSP カーネルを活用し始めている [3, 4])。

ただしその設計にあたっては、産業界での応用も念頭に置き、それに堪える性能を実現することも目指した。また、 μ ITRON 仕様 OS 用のミドルウェア開発のための標準プラットフォームとしての利用や、 μ ITRON 仕様 OS を自分で実装しようという人のためのリファレンス実装としての利用も想定している。特に、ミドルウェア開発のためのプラットフォームとしての利用を考えると、スタンダードプロファイルに対して拡張を加えていないことが重要になる。つまり、TOPPERS/JSP カーネルで動作が確認されたミドルウェアは、スタンダードプロファイル以上の機能を持った他の μ ITRON4.0 仕様 OS へも容易にポータリングできることになる。

TOPPERS/JSP カーネルは、オープンソースのフリーソフトウェアとして、ウェブサイト [5] からソースコードの形で配付している。組込みシステムの特性を考慮して、TOPPERS/JSP カーネルに対しては、フリーソフトウェアの中でも特に柔軟な利用条件を設定した。これにより、利用条件の面でも産業界での応用に堪えるものと考えている。

なお、「TOPPERS」は、「Toyohashi OPen Platform for Embedded Real-time Systems」の略称であり²、組込みリアルタイムシステム構築の基盤となる各種のソフトウェア設計資産の開発を行うことを目的に、豊橋技術科学大学 組込みリアルタイムシ

²ちなみに、topper という英単語には、口語で「すぐれたもの」「傑作」などの意味がある。

ステム研究室 (高田研究室) を中心に進めているプロジェクトの名称である。

3 開発の経緯

応募代表者 (高田) は、数年前から、ItIs (ITRON Implementation by Sakamura Lab.) という名称の ITRON 仕様に準拠したリアルタイム OS を開発し、フリーソフトウェアとして配付してきた。ItIs は、TOPPERS/JSP カーネルと同様、研究・教育への利用を一義的な目的として開発したもので、実際、リアルタイム OS に関する数々の研究・教育活動に活用してきた (例えば、[6] や [7])。また、産業界においても、ミドルウェア開発のためのプラットフォームや、 μ ITRON3.0 仕様のリファレンス実装、情報機器のプロトタイプ開発環境として利用された。さらには、開発時には全く想定していなかったが、実際の機器に組み込む形で利用された。

μ ITRON4.0 仕様の策定作業は、1997 年頃から応募代表者が中心になって進めてきたが、1999 年 6 月には策定作業が完了し、仕様書を公開することになった。完成した μ ITRON4.0 仕様を普及させるためには、リファレンス実装があることが望ましいし、また策定した仕様の有効性を評価するためにも、 μ ITRON4.0 仕様に準拠したリアルタイム OS を開発する必要があると考えた。

そのためには、ItIs を改造して μ ITRON4.0 仕様に準拠させる方法もあったが、以下に述べるような理由により、TOPPERS/JSP カーネルという名称で、スクラッチから開発することにした (もちろん、ItIs の開発で得られた知験は盛り込まれている)。

- ItIs の開発時には、実際の機器に組み込んで利用することまでは想定していなかったため、実行時性能よりも、読みやすさや改造しやすさを重視して実装した。そのため、製品化されている μ ITRON 仕様 OS と比べて、実行時性能の面ではかなり見劣りするものであった。TOPPERS/JSP カーネルでは、機能をスタンダードプロファイルに絞り込むことで、実行時性能と読みやすさ・改造しやすさの両立を目指すこととした。
- ItIs には、 μ ITRON 仕様に対する各種の拡張機能を盛り込んだため、ソースコード中に条件コンパイル指定 (#if や #ifdef) が多数含まれることになり、リアルタイム OS の初心者にとっては、非常に読みにくいものであった (上級者にとっては、読みやすかった)。この点についても、機能をスタンダードプロファイル「ちょうど」と決めることで、ほとんどの条件コンパイル指定を取り除くことができ、初心者にも読みやすいもの

とすることを旨とした。

- ItIs は、当初、トロン仕様のマイクロプロセッサをターゲットとして開発され、トロン仕様プロセッサの持つ OS 向けの特殊な機能を活用していたため、他のプロセッサにポータリングした場合に効率が悪いものとなっていた。TOPPERS/JSP カーネルでは、より一般的なプロセッサで効率が出るような構造を採用することとした。また、プロセッサの抽象化レベルを上げることで、異なるプロセッサへのポータリングが容易になるよう考慮した。

以上の経緯より、1999 年末頃に TOPPERS/JSP カーネルの開発に着手し、2000 年 11 月にそのバージョン 1 をリリースした。

オープンソースの μ ITRON 仕様 OS に対する反響は予想以上のもので、リリースしてからの約 2 週間で、約 600 のサイトにダウンロードされた。また、ユーザのためのメーリングリストへの登録者も約 80 名となり (この数は、すでに ItIs ユーザのためのメーリングリストへの登録者数を越えている)、ポータリング作業や改良を行うユーザも出てきている。

4 TOPPERS/JSP カーネルの機能

2節で述べた通り、TOPPERS/JSP カーネルでは、 μ ITRON4.0 仕様のスタンダードプロファイル「ちょうど」の機能を実装することを原則としたが、後述の理由で若干の拡張機能も実装している。

具体的には、スタンダードプロファイルに含まれる 72 のサービスコールと 11 の静的 API のすべてに加えて、割込みの禁止・許可のための 4 つのサービスコール、CPU 例外発生時のシステム状態参照のための 5 つのサービスコール、性能評価のためのシステム時刻を参照するサービスコール (1 つ) をサポートしている。表 1 に、TOPPERS/JSP カーネルの主な機能を示す。

この中で、CPU 例外発生時のシステム状態参照のための機能は、スタンダードプロファイルでサポートが必要とされているが、具体的なサポート方法が規定されていないものであり、拡張機能ではない。他の 5 つのサービスコールが、スタンダードプロファイルからの拡張機能ということになる。スタンダードプロファイル「ちょうど」の原則にもかかわらずこれらの拡張機能をサポートすることとしたのは、割込みの禁止・許可のための 4 つのサービスコールについては実装方法の妥当性の検証と実用性を、性能評価のためのシステム時刻参照について研究目的で利用する際の利便性を重視したためである。

TOPPERS/JSP カーネルがサポートしているサ

1. タスク管理機能
 - タスクの生成 (静的 API)
 - タスクの起動と終了
 - タスクの優先度の参照と変更
2. タスク付属同期機能
 - タスクの起床待ちと起床
 - タスクの待ち状態の強制解除
 - タスクの強制待ち状態への移行と再開
 - タスクの遅延
3. タスク例外処理機能
 - タスク例外処理ルーチンの定義 (静的 API)
 - タスク例外の要求
 - タスク例外の禁止と許可
4. 同期・通信機能
 - セマフォ機能
 - イベントフラグ機能
 - データキュー機能
 - メールボックス機能
5. メモリプール管理機能
 - 固定長メモリプール管理機能
6. 時間管理機能
 - システム時刻管理機能
 - 周期ハンドラ機能
 - 性能評価用システム時刻参照機能
7. システム状態管理機能
 - タスクの優先順位の変更
 - 実行状態のタスクの参照
 - CPU ロック状態への移行と解除
 - ディスパッチの禁止と許可
 - システム状態の参照
 - CPU 例外発生時のシステム状態参照
8. 割り込み管理機能
 - 割り込みハンドラの定義 (静的 API)
 - 割り込みの禁止と許可
9. システム構成管理機能
 - CPU 例外ハンドラの定義 (静的 API)
 - 初期化ルーチンの追加 (静的 API)

表 1: TOPPERS/JSP カーネルの主な機能

ービスコールおよび静的 API の一覧については、ユーザズマニュアル [8] の 8 章を参照されたい。

5 動作環境と使用したツール

2節で述べた通り、TOPPERS/JSP カーネルは、ターゲットプロセッサとして、現時点で、モトローラ社の M68040 と日立製作所の SH3 をサポートしている。開発に用いているボードは表 2 の通りであり、これらのボード上でのカーネルの動作を確認している。

TOPPERS/JSP カーネル自身の開発には、ホスト環境用のツールとして、標準規格に準拠した C コンパイラと C ライブラリ、標準規格に準拠した C++ コンパイラと C++ ライブラリ (STL を含む)、perl (バージョン 5.005 で動作確認)、GNU make (バージョン 3.77 で動作確認) が、クロス環境用のツールとして、GNU C コンパイラ (GCC: バージョン 2.95.2 で動作確認) と GNU のバイナリユーティリティ (BINUTILS: バージョン 2.10 で動作確認) が必要である。BINUTILS には、アセンブラやリンカ、シンボリックデバッガなどが含まれている。また、TOPPERS/JSP カーネル上で動作するアプリケーションソフトウェアを開発するには、上記に加えて、標準 C ライブラリがあることが望ましい。標準 C ライブラリについては、GNU 開発環境の中の NEWLIB と呼ばれるライブラリを利用した (バージョン 1.8.1 で動作確認)。これらのツールは、すべてフリーのソフトウェア開発環境であるため、カーネル本体のみならずツールまでもフリーで入手し、ソフトウェア開発をおこなうことが可能である。

Linux のシミュレーション環境は、バージョン 2.2 以上のカーネルと、バージョン 2.1 以上の glibc (標準ライブラリ) で動作する。実際に動作確認したディストリビューションは、Vine Linux 2.0 と Debian GNU/Linux 2.1 である。用いるソフトウェア開発環境は、上と同様である。

また、Windows 上のシミュレーション環境は、95/98/NT4/2000 上で動作する (ただし、一部の機能は Windows NT4/2000 のみに対応)。ソフトウェア開発環境としては、Visual C++ 6.0 を用いた。

6 TOPPERS/JSP カーネルの特徴

TOPPERS/JSP カーネルは、研究・教育へ利用するために開発したものであるが、その開発自身は研究テーマとは位置付けていない。そのため、TOPPERS/JSP カーネルの実装手法には、取り立てて主張するような新規な点はない。むしろ、教育への利用を考えて、オーソドックスな方法で実装されている。

ただし、以下で説明するように、ターゲット独立部とターゲット依存部の分離方法については、他のターゲットプロセッサへのポーティングの容易性と高い実行時性能を両立させるために、工夫を凝らしている。また、読みやすさ・改造しやすさと高い実行時性能の両立にも、注意を払っている。

すなわち、TOPPERS/JSP カーネルの最大の新規性は、ポーティングしやすさ、読みやすさ、改造しやすさを高いレベルで維持しつつ、製品化されている μ ITRON 仕様 OS と比べても遜色のない性能

プロセッサ (型番)	ボード (ボードメーカー)
M68040 (MC68LC040)	DVE-68K/40 (電産)
SH3 (7709A)	SH-CARD CARD-E09A (セイコーエプソン)
SH3 (7709)	MU-200-RSH3 (三菱電機マイコン機器ソフトウェア)
SH3 (7708)	DVE-SH7700 (電産)

表 2: 開発に用いたボード

を実現していることであると考えている。ポーティングしやすさについては、他のプロセッサへのポーティングを 3 日間で完了したという報告がある (以下で詳述する)。残念ながら、読みやすさや改造しやすさを定量的に評価することは難しく、第三者による評価を待ちたいと考えている³。

以下では、TOPPERS/JSP カーネルの主な特徴を説明する。

- 読みやすく改造しやすいソースコード

上述の通り、TOPPERS/JSP カーネルは、ソースコードの読みやすさや改造しやすさに重点を置いて実装した。ただし、安易な読みやすさを追求して、効率の悪い平易なアルゴリズムを採用することはしていない。むしろ、タイムイベントの管理にヒープ構造を用いるなど⁴、複雑であっても効率的なアルゴリズムは積極的に採用した。また、カーネルの理解を容易にするために、カーネル設計時の考察点を記述した設計メモ [9] も公開している。

- 他のターゲットへのポーティングが容易な構造

カーネルのできる限り多くの部分を C 言語で記述する、ターゲット独立部とターゲット依存部を明確に分離するなど、他のターゲットプロセッサやボードへのポーティングが容易な構造とした。特に割込みの処理は、実行時性能を向上させる上で非常に重要なポイントである一方で、プロセッサによる違いが大きく、安易に隠蔽すると実行時性能の低下につながる。

また、ユーザがポーティングすることを容易にするために、ターゲット依存部のインタフェース仕様 [10] も公開している。その結果、TOPPERS/JSP カーネルを最初にリリースした 1 週間後には、ユーザによって i386 へポーティングされたものが公開された。ポーティングを行ったユー

³すでに何人かのリアルタイム OS 技術者から、読みやすいという評価をいただいている。

⁴ほとんどのリアルタイム OS で採用されているキュー構造で実装した場合、いずれかの処理で、タイムイベントの数 n に対して $O(n)$ の時間がかかることは避けられない。それに対して、ヒープ構造を用いると、すべての処理を $O(\log n)$ で行うことができる。

測定条件	タスク切替え時間
DVE-68K/40	
キャッシュON	13 μ 秒
キャッシュOFF	32 μ 秒
CARD-E09A	
ライトバックキャッシュ	2 μ 秒
ライトスルーキャッシュ	6 μ 秒
キャッシュOFF	43 μ 秒

表 3: タスク切替え時間

ザからは、ポーティング作業は 3 日間でほぼ完了したと報告されている。また、応募者の一人 (若林) が行った V850 へのポーティングも、実質 3 日程度で完了している⁵。

- 高い実行性能と小さい RAM 使用量

大部分が C 言語で記述されているカーネルとしては、高い実行性能と小さい RAM 使用量を実現している。

表 3 に、DVE-68K/40 と SH-CARD CARD-E09A を用いて、いくつかの条件下で測定した rot_rdq サービスコールによるタスク切替え時間を示す。DVE-68K/40 は、MC68LC040 (MC68040 の FPU を持たないタイプ) をバスクロック 33MHz (内部クロックは 66MHz) で動作させている。CARD-E09A のデータは、内部クロック 133MHz、バスクロック 33MHz で動作させた場合の数値である⁶。

RAM 使用量を示すデータとしては、タスク毎に必要な制御ブロックのサイズは、RAM 上に置く必要のあるタスク制御ブロック (TCB) が 32 バイト、ROM 上に置くことができるタスク初期化ブロックが 32 バイトとなっている (タスクの動作には、この他にスタック領域が必要である)。現バージョンでは、ROM の使用量はそれほど重視

⁵3 日間という期間は、ボードサポート (シリアルインタフェースドライバなど) のポーティング作業は含まず、プロセッサについて熟知しているという前提での期間である。実際には、プロセッサの機能を理解する方が、ポーティングよりも長い期間がかかると思われる。

⁶CARD-E09A でキャッシュOFF 時の数値が悪いのは、メモリが遅いことが原因となっている。

しておらず、タスク初期化ブロックのサイズは、まだ最適化の余地がある。

- Linux および Windows 上でのシミュレーション環境を用意

Linux および Windows 上でのシミュレーション環境は、Linux および Windows の 1 つのプロセスの中で、複数のタスクを切り替えて動作させることで、TOPPERS/JSP カーネルの動作をシミュレートするものである。

このようなシミュレーション環境を用いると、ハードウェアが完成する前にソフトウェアの論理検証が行えるため、組込みソフトウェアのプロトタイプ開発に用いられることが多くなっている。また、ハードウェア的にはパソコンのみで動作するために、リアルタイム OS の学習用途にも最適なものである。

- フリーソフトウェアのみで開発環境まで構築可能
前述した通り、TOPPERS/JSP カーネル自身とその上で動作するアプリケーションソフトウェアの開発環境を、すべてのフリーのソフトウェアで構築できる。これは、研究・教育機関で広く利用してもらうためには、重要な要件である。また、TOPPERS/JSP カーネルの動作するターゲットシステムを、GNU のデバッガ (GDB) を用いてリモートデバッグするために必要となるデバッグモニタ (これを、GDB では stub と呼んでいる) についても、GDB と一緒に配付されているものをベースに、TOPPERS/JSP カーネルと共用できるよう改造を行った。

7 今後の計画

我々は、今後も TOPPERS/JSP カーネルの完成度を上げる作業を進めるとともに、他のプロセスへのポーティングやシミュレーション環境の充実、ITRON デバッグインタフェース仕様に準拠したデバッグ環境の構築など、その適用性を広げるための改良やサポート環境の充実を行っていくことを予定している。また、TOPPERS/JSP カーネルを題材にした教材の開発にも力を入れていきたいと考えている。

TOPPERS プロジェクトでは、TOPPERS/JSP カーネルを手始めとして、組込みリアルタイムシステム構築の基盤となる各種のソフトウェア設計資産の開発を進めていく計画である。そのためには、オープンソースの利点を活用し、プロジェクトへの参加を希望する組織の協力を得て進めていきたいと考えている。すでに、宮城県産業技術総合センターなど、複数の大学ならびに研究機関がプロジェクトへの参加を検討している。

さらに、TOPPERS/JSP カーネルが産業界で広く使われるためには、単にオープンソースのフリーソフトウェアというだけでは不十分で、TOPPERS/JSP カーネルに対するサポートビジネスやインテグレーションビジネスを行う企業や、TOPPERS/JSP カーネル上で動作するミドルウェアを開発・販売する企業が出てくる必要があると考えられる。そのようなビジネスを行おうとする企業が出てきた際には⁷、大学としても可能な支援を行いたいと考えている。

参考文献

- [1] ITRON プロジェクトホームページ,
<http://www.itron.gr.jp/>.
- [2] 高田 広章 編, μ ITRON4.0 仕様書, トロン協会, 1999 年 6 月.
- [3] 若林隆行, 高田広章, “ITRON デバッグインタフェース仕様の概要とその適応性に関する評価,” 情報処理学会シンポジウムシリーズ (コンピュータシステム・シンポジウム論文集), vol. 2000, no. 13, pp. 53–60, 2000 年 11 月.
- [4] 本田晋也, 高田広章, “ μ ITRON4.0 仕様における例外処理機能とその評価” (投稿予定).
- [5] TOPPERS/JSP カーネル ウェブサイト,
<http://www.ertl.ics.tut.ac.jp/TOPPERS/>.
- [6] H. Takada and K. Sakamura, “Experimental implementations of priority inheritance semaphore on ITRON-specification kernel,” Proc. 11th TRON Project Symposium, pp. 106–113, IEEE CS Press, 1994 年 12 月.
- [7] H. Takada and K. Sakamura, “A novel approach to multiprogrammed multiprocessor synchronization for real-time kernels,” Proc. 18th IEEE Real-Time Systems Symposium (RTSS), pp. 134–143, 1997 年 12 月.
- [8] 豊橋技術科学大学 組込みリアルタイムシステム研究室, TOPPERS/JSP カーネルユーザズマニュアル, 2000 年 11 月.
- [9] 豊橋技術科学大学 組込みリアルタイムシステム研究室, JSP カーネル 設計メモ, 2000 年 11 月.
- [10] 豊橋技術科学大学 組込みリアルタイムシステム研究室, JSP カーネル ターゲット依存部 インタフェース仕様, 2000 年 11 月.

⁷すでに、TOPPERS/JSP カーネルのサポートを表明している企業がある。