



ITRONプロジェクトにおける 標準化活動の最新状況

2000年11月17日

高田 広章

豊橋技術科学大学 / (社)トロン協会 ITRON部会

hiro@ertl.ics.tut.ac.jp

ITRONプロジェクトホームページ

<http://www.itron.gr.jp/>

ITRONプロジェクト



- ▶ TRONプロジェクトのサブプロジェクトの1つ

目的

- ▶ 組み込みシステム用のリアルタイムOSとそれに関連する仕様の標準化を行う

第1フェーズ(1984年～)

- ▶ リアルタイムカーネル仕様の標準化に注力

リアルタイムカーネル

= リアルタイムOSの核になるモジュール

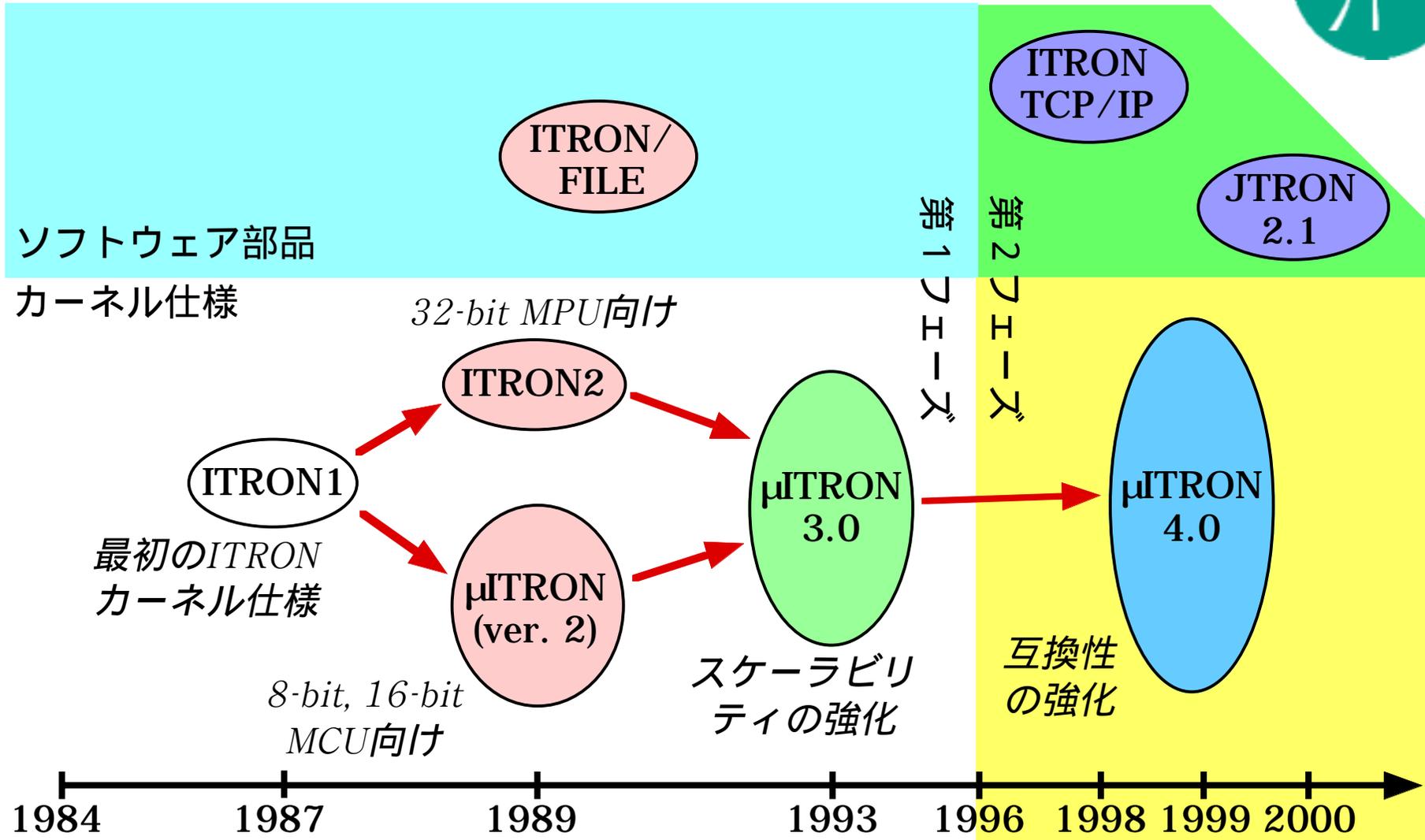
背景：小規模な組み込みシステムではカーネルの機能しか
必要ないケースが多い

第2フェーズ(1996年頃～)

- ▶ 周辺仕様まで含めた標準化へ

背景：組み込みシステムの大規模化・複合化

ITRON仕様の歴史



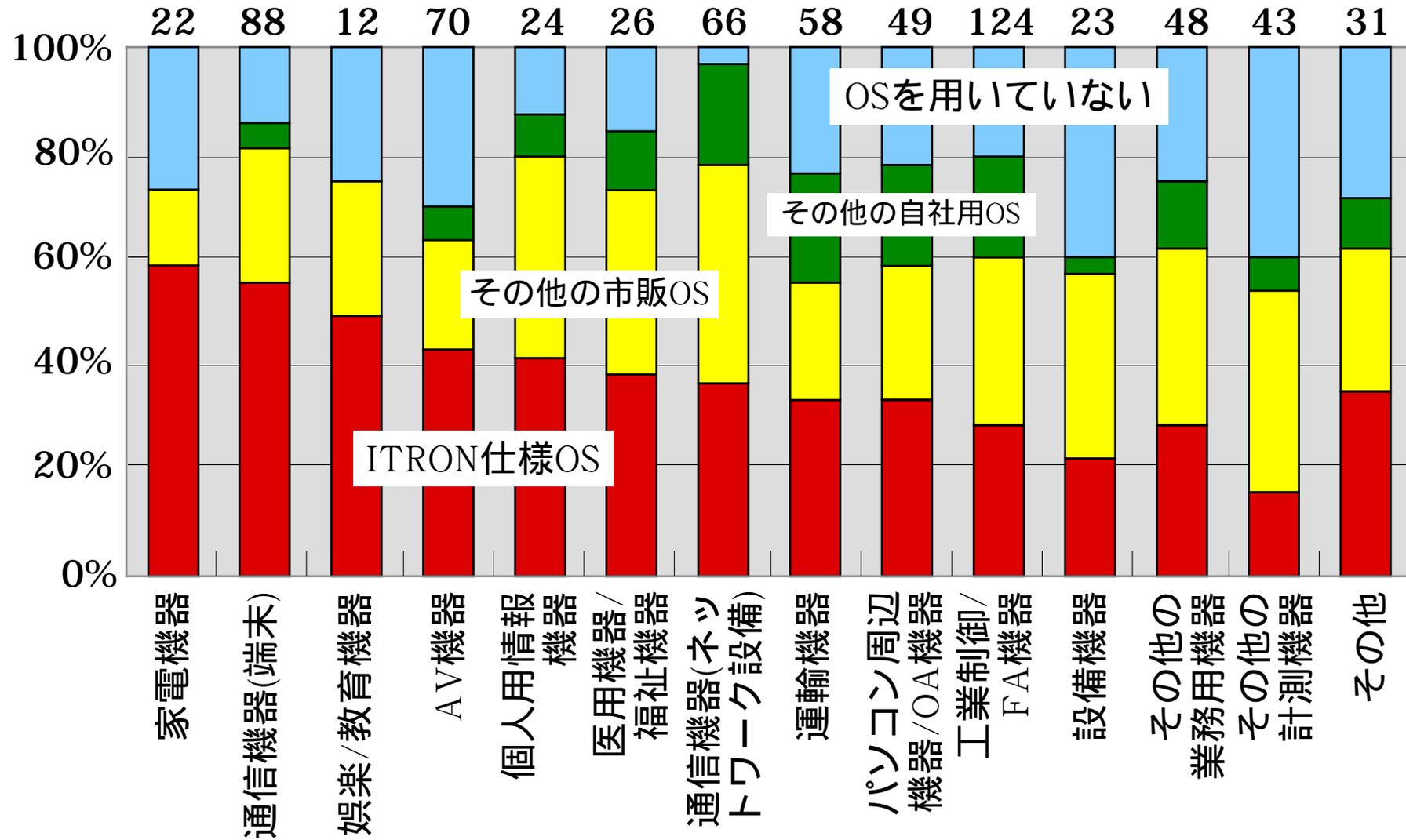
ITRON仕様カーネルの特徴



- ▶ OSの小型軽量化が可能
 - ▶ ワンチップマイコンにも適用可能
- ▶ 仕様の理解が容易
 - ! 技術者教育のための標準化の側面を重視
- ▶ 完全にオープンな標準仕様
 - ▶ ロイヤリティなしで実装することができる
- ▶ 多種多様なプロセッサ用に実装できる/されている
 - ▶ 8-bitワンチップマイコンから32-bit RISCマイコンまで
 - ▶ 異なるプロセッサへの移行が容易に
- ▶ 多くの機器で使用実績がある
 - ▶ 組込みシステム分野で最も広く使われているOS仕様
- ▶ 多くのメーカー/ベンダがサポート

ITRON仕様カーネルの利用状況

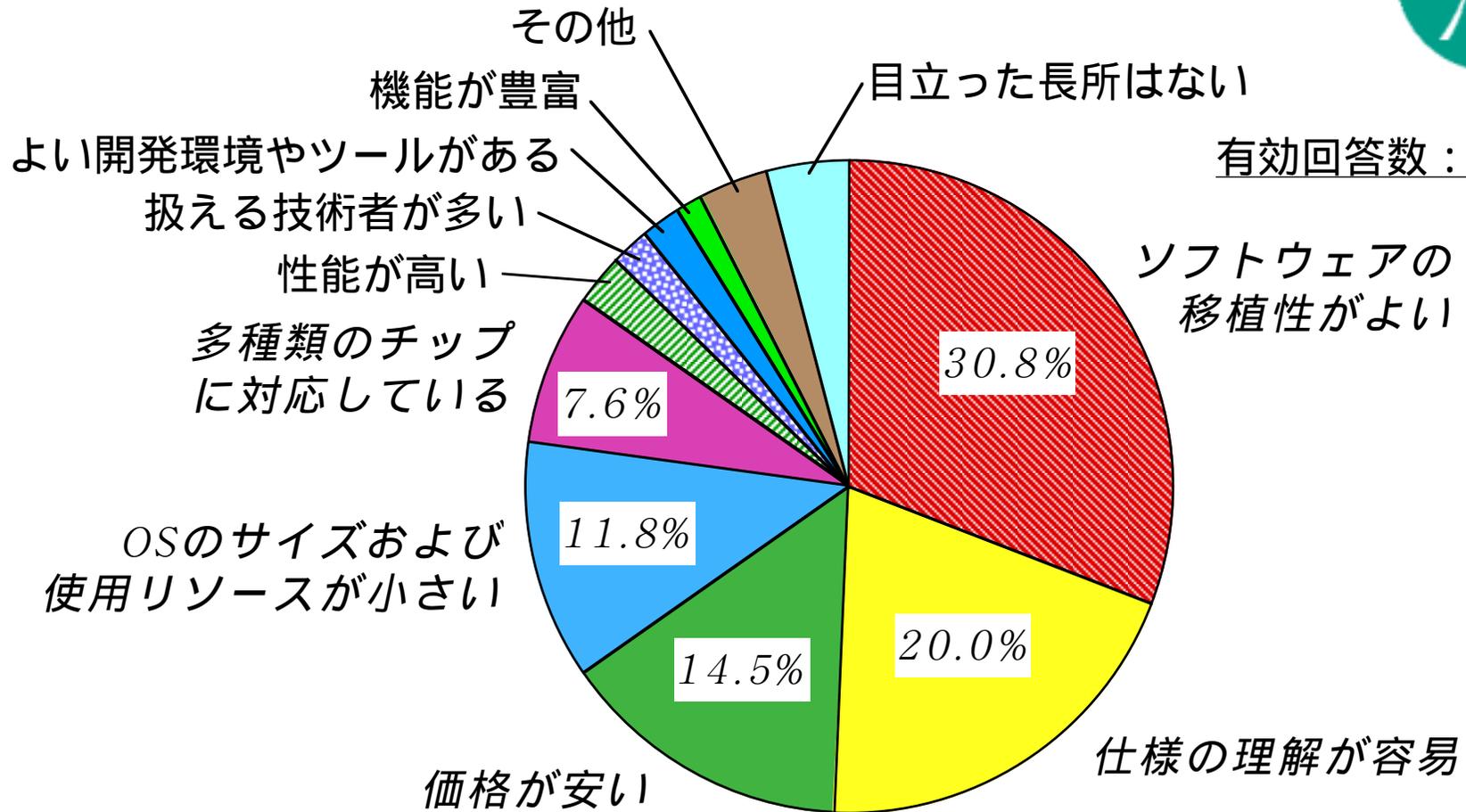
有効回答数：684



ITRON仕様に対するユーザからの評価

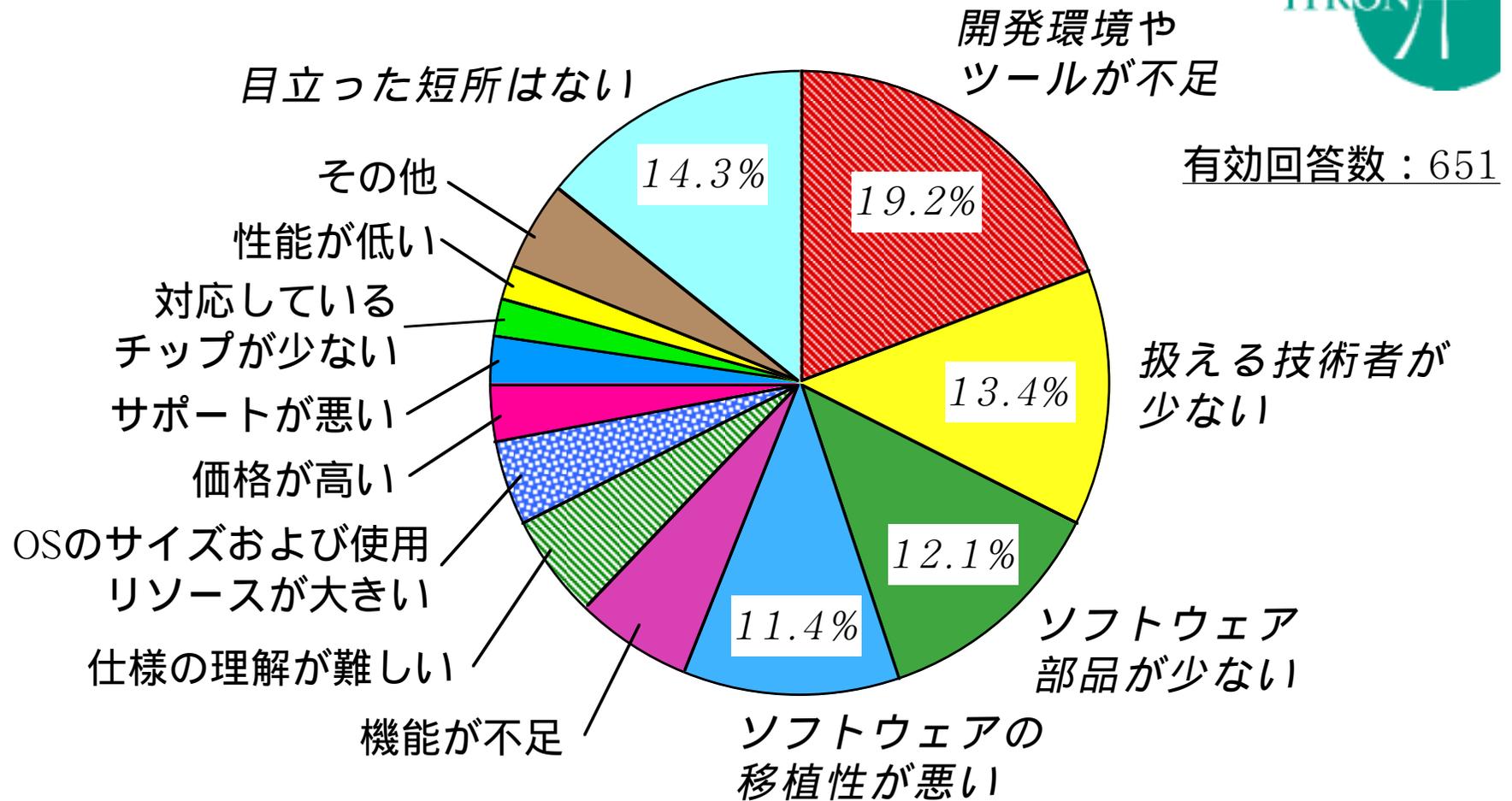


有効回答数：638



ITRON仕様OSの長所

(トロン協会による調査, 1999年末～2000年頭, 日本)



ITRON仕様OSの短所

(トロン協会による調査, 1999年末～2000年頭, 日本)

ITRONプロジェクト - 第2フェーズ



周辺仕様まで含めた標準化へ

カーネルとの関連での標準化

→ 周辺仕様そのものの標準化

ソフトウェア部品 (ソフトウェアIP, 実行時ソフトウェア)

- ▶ ソフトウェア部品が流通する前提条件の整備
- ▶ ソフトウェア部品の API の標準化 (部品の種類毎)

開発環境・言語

- ▶ カーネルとデバッグ環境間のインタフェースの標準化
- ▶ C言語以外のプログラミング言語バインディング

応用分野に特化した標準化

- ▶ 応用分野に固有の要求への対応



第2フェーズの標準化活動の状況整理

ソフトウェア部品が流通する前提条件の整備

- ▶ μ ITRON4.0仕様 1999年6月公開
- ▶ μ ITRON4.0仕様 検定仕様書 検討中

ソフトウェア部品のAPIの標準化

- ▶ ITRON TCP/IP API仕様 1998年5月公開
- ▶ JTRON2.0仕様 → JTRON2.1仕様 2000年11月公開
- ▶ デバイスドライバ設計ガイドライン 中間報告を公開
- ▶ TRON GUI仕様 検討中

カーネルとデバッグ環境間のインタフェースの標準化

- ▶ ITRONデバッグインタフェース仕様 暫定仕様書を公開

C言語以外のプログラミング言語バインディング

- ▶ C++ / EC++言語バインディングの標準化 方向性見直し

応用分野に固有の要求への対応

- ▶ 自動車制御応用における要求の整理 成果を μ ITRON4.0仕様へ反映

μITRON4.0仕様



- ▶ 最新の μITRONリアルタイムカーネル仕様
- ▶ 1999年6月に仕様をフィックス・公開

新仕様策定の目的

- ▶ ソフトウェアの移植性の向上
 - ▶ 組込みソフトウェアの大規模化により移植性が重視
 - ▶ 移植性の向上はソフトウェア部品流通の前提条件
- ▶ ソフトウェア部品向け機能の追加
 - ▶ 外販することを前提としたソフトウェア部品開発
- ▶ 新しい要求・検討成果の反映
 - ▶ リアルタイム性の保証を容易にするための機構
 - ▶ よりコンパクトな実装を可能にする仕様
- ▶ 半導体技術の進歩への対応



μITRON4.0仕様のフォローアップ

- ▶ 英文版の作成作業を続行中
- ▶ 検定仕様書を作成中
- ▶ JTRON仕様をμITRON4.0仕様に整合させた
- ▶ 細かな問題点を整理中 → Ver. 4.01 へ

μITRON4.0仕様準拠製品（トロン協会に登録されているもの）

- ▶ PrKERNEL v4（エルグ）
- ▶ μMORE v4.0（ACCESS）
- ▶ Nucleus μIPlus（グレープシステム / ATI）

μITRON4.0仕様準拠のフリーソフトウェア

- ▶ TOPPERS/JSPカーネル（豊橋技科大）
<http://www.ertl.ics.tut.ac.jp/TOPPERS/>

μITRON4.0仕様 検定仕様書



動機とこれまでの経緯

- ▶ ITRON仕様に従って実装されたカーネルが，本当に仕様を満たしているか確認手段が欲しい
 - ▶ ユーザの立場からは，何らかの認証制度があると判断が容易にでき，採用にあたって安心
 - ▶ 「堅い」認証制度は高いコストがかかり，ITRONにはなじまないとの考え方
- ! 弱い標準化により，認証作業は技術的にも困難
 - これまで，ITRON仕様には認証制度を設けていなかった(代わりに登録制度を設けていた)
- ! μITRON4.0仕様のプロファイル規定は「強い標準化」であり，認証作業の技術的な問題はない



μITRON4.0仕様の検定制度の考え方 (検討中)

! このページの内容は今後の検討によって変わる可能性があります

- ▶ なるべく簡便な検定制度とする
 - ▶ 仕様に合致していることをテストしたという自己申告を以て、検定合格とする
 - ▶ 後で要請があれば、テストした証拠を提出する義務
 - ▶ μITRON4.0仕様のスタンダードプロファイルと自動車制御用プロファイルを対象



- ▶ どのようなテストをしなければならないかの規定 (つまり、テスト仕様書) の策定が必要



- ▶ μITRON4.0仕様 検定仕様書



検定仕様の考え方

- ▶ 検定のためのテストは，仕様書を間違えて実装していないかを確認するための最低限のテスト
 - ! 検定のためのテストは，信頼性を上げる (= バグをなくす) ためのテストではない



- ▶ 仕様書に沿ったテストを行う
 - ▶ 仕様書で1箇所に書かれていることは，テストも1回
 - ▶ 網羅的な組み合わせテストは行わない
 - ▶ マニュアルやヘッダファイルの検査も実施

検定仕様書の構成

- ▶ 検定仕様書の目的と位置付け，作成にあたっての考え方
- ▶ テスト項目 (スタンダード / 自動車制御用プロファイル)
- ▶ テスト手順 (スタンダード / 自動車制御用プロファイル)



テスト項目

- ▶ 仕様書に沿ってテストすべき項目を列挙

例) act_tsk-3

対象タスクが休止状態である場合には、休止状態から実行可能状態に移行させること。

- ▶ そのテスト項目を実施するテスト手順 (一つ) を示す

テスト手順

- ▶ テスト項目を網羅するようなテストの手順を提示
 - ▶ どのタスクからどのサービスコールを呼び出し、どのような結果になるべきかというレベルで記述
- ▶ そのテスト手順で実施するテスト項目 (複数) を示す

作成作業の状況

- ▶ テスト項目の数え上げとテスト手順の設計がほぼ完了

デバッグ環境とのインタフェースの標準化



開発環境やツールが不足する背景

- ▶ ITRON仕様OSの場合，カーネルメーカーと開発環境メーカーが別の会社であるケースが多い
- ▶ 数多くのITRON仕様カーネル
- ▶ ITRON仕様はデバッグサポート機能を規定していない
- ▶ ITRON仕様はAPIのみを定めており，その実装方法は実装者に任されている



現状の問題点

- ▶ デバッグツール (デバッガやICEなど) はそれぞれのITRON仕様カーネルに個別に対応することが必要
- ▶ 個々のITRON仕様カーネルのシェアは小さく，デバッグツールメーカーのサポートが後回しになる



デバッグインタフェースの標準化

- ▶ ITRON仕様のリアルタイムカーネルとデバッグツールの**RTOSサポート機能**とのインタフェースを標準化
- ▶ デバッグツールのRTOSサポート機能とは...
 - ▶ RTOSオブジェクトの状態読出し
 - ▶ タスクのコンテキストの読出しと変更
 - ▶ RTOSのサービスコールの発行
 - ▶ RTOSに関わるブレークポイントの設定
 - ▶ RTOSの実行履歴の取得・表示

標準化により...

- ▶ デバッグツールが，多種のITRON仕様カーネルへ対応することが容易に
- ▶ カーネル側は，開発環境を充実させることが容易に



標準化の前提

- ▶ RTOSサポート機能を持たないデバッグ環境は実現できている
- ▶ 開発ホストは十分な能力を持っている

標準化の目標

- ▶ スケーラブルなインタフェース仕様
 - ▶ 小規模なターゲットシステムの場合には，ターゲットシステムのオーバヘッドを最小限に
 - ▶ 大規模なターゲットシステムで，少々のオーバヘッドが許される場合には，デバッグ作業の高速化を可能に
- ▶ 異なるデバッグツール (例: デバッガとICE) とのインタフェースをできる限り共通化
- ▶ 他のRTOSやソフトウェア部品にも (少なくとも，考え方やアーキテクチャは) 適用可能



よくあるアプローチ

- ▶ カーネルにデバッグサポート機能を持たせる
 - ! ITRON2仕様策定時にデバッグ用のAPIを検討したことあり
 - ➔ ターゲットシステムにかかるオーバーヘッドが大き
く、小規模なターゲットシステムには適用困難
- ▶ カーネルの内部構造を標準化する
 - ➔ カーネル実装時の工夫の余地を減らし、ハードウエ
アや応用への適応性を下げる
 - ➔ いずれの方法も、既存のカーネルの改造が必要

工夫したアプローチ

- ▶ カーネルの内部構造を記述する言語を標準化する

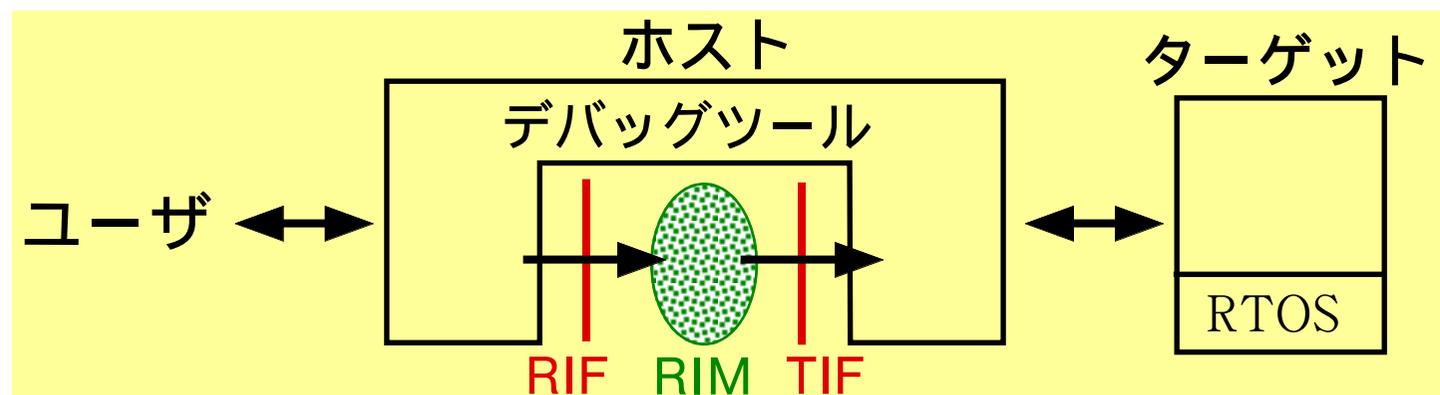
さらに...

- ▶ カーネルの内部構造にアクセスするプログラムモジュールを標準化する



アーキテクチャ

- ▶ RTOSインタフェースモジュール (RIM)
 - ▶ あるカーネルに対応して、デバッグツール側からの処理要求を、デバッグツールのプリミティブを使って処理する方法を、カーネルメーカーが**プログラムの形**で提供
- ▶ RTOSアクセスインタフェース (RIF)
 - ▶ デバッグツールからRIMに対して処理を要求するインタフェース
例) あるオブジェクトの状態を読み出す
- ▶ ターゲットアクセスインタフェース (TIF)
 - ▶ RIMがデバッグツールのプリミティブを呼び出すインタフェース
例) 指定した番地のメモリの内容を読み出す





デバッグインタフェース仕様の規定内容

- ▶ RTOSアクセスインタフェース (RIF)
- ▶ ターゲットアクセスインタフェース (TIF)
 - ▶ いずれもC言語の関数インタフェースの形で規定

ガイドライン・その他の規定

- ▶ RTOSサポート機能ガイドライン
 - ▶ デバッグツールがRTOSをサポートしているとは何を意味しているかを定義．ユーザに機能の誤解を与えないように，言葉使いを標準化することが重要
- ▶ Windows DLL規定
 - ▶ RIMをWindowsのDLLとして提供する場合の追加規定
 - ▶ これに準拠することで，RIMをバイナリ形式で提供できる
- ▶ 動作履歴ファイル形式の規定
 - ▶ RTOSの動作履歴をファイルに保存する場合の形式の規定
 - ▶ 独立した動作履歴解析ツールが可能に



標準化活動の状況

- ▶ 暫定仕様書を公開中
- ▶ 暫定仕様書に対する意見を募集中
debug-wg@itron.gr.jp
- ▶ 暫定仕様書の評価実装を実施中
- ▶ 暫定仕様をベースに実装を開始したメーカーも(?)

OSEK/VDXプロジェクトとの共同標準化

- ▶ 昨年，OSEK/VDXプロジェクトに対して，デバッグインターフェース仕様の共同標準化を提案
 - ! OSEK/VDX仕様OSにもITRON仕様OSと類似の状況がある
- ↓
- ▶ OSEK/VDXプロジェクトの Debug Interface WG が活動を開始したが，すぐに共通化することは難しい状況

JCGプロジェクト



- ▶ μ ITRON仕様カーネル上で動作するソフトウェア
ア部品の整備を図る
 - ➔ 情報家電のための分散プラットフォーム
 - ▶ JTRON2.0仕様（Java VM は既製品を利用）
 - ▶ 組み込みシステム用の CORBA
 - ▶ 組み込みシステム用の GUI パッケージ
 - ➔ フリーソフトウェアとして公開
- ▶ 情報処理振興事業協会（IPA）の「次世代デジタル応用
基盤技術開発事業」の採択テーマの1つ

ITRONプロジェクト 今後の計画



- ▶ 組み込みシステム開発のオープン化の流れ



既製の技術を安心して使うには標準化が不可欠

μITRON4.0仕様はオープン化の流れに対応する基盤

1998年

ITRON TCP/IP API仕様

JTRON仕様

要求事項

整合性

2001年

次の展開

2000年 足場固め中

デバッグイン

タフェース仕様

C++ API仕様

検定仕様書

μITRON4.0仕様

1999年 足場ができた

Call for Contributions

- ▶ ITRON仕様の検討作業の多くの部分は，オープン (貢献する意志があれば誰でも参加可能) な研究会で行っている