



# ITRON仕様の最近の現状

2001年2月

(社)トロン協会 ITRON部会

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

## リアルタイムOS仕様の標準化



### 仕様の標準化のメリット

- ▶ ソフトウェアの再利用が容易に
- ▶ ソフトウェア技術者の教育が容易に
- ▶ 品質の高いOS製品がリーズナブルなコストで入手可能に

### 標準化が可能な背景

- ▶ 現世代のリアルタイムOS技術は competitive ではない
- ▶ 多くの組み込みシステムにおいては、どのようなリアルタイムOSを用いるかが製品の価値を決めるわけではない
  - リアルタイムOSは単なる部品の一つ

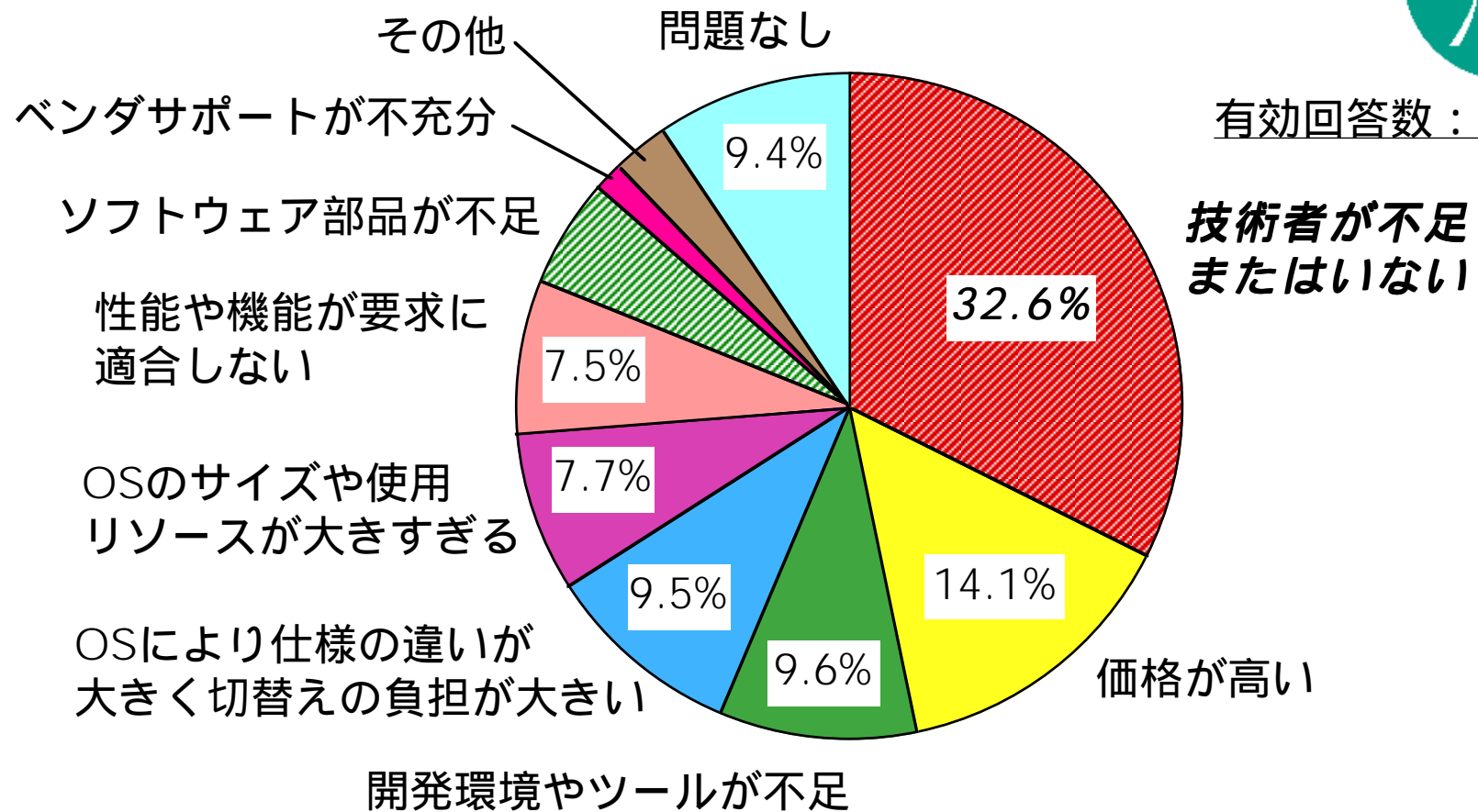
### ITRON以外の標準リアルタイムOS仕様

- ▶ POSIX とそのリアルタイム拡張
- ▶ OSEK/VDX OS仕様



# リアルタイムOS使用上の問題点

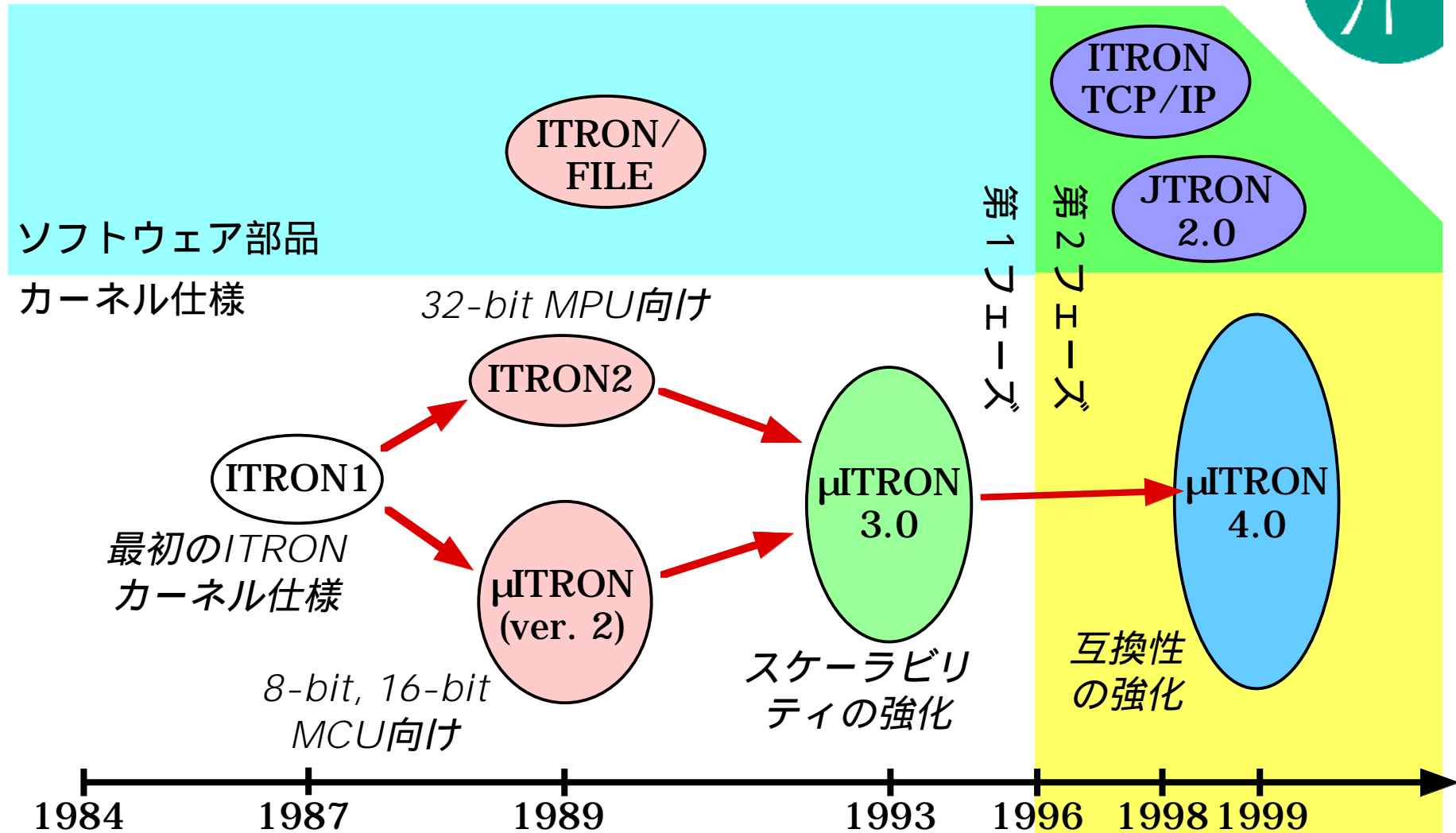
有効回答数：757



## リアルタイムOS使用上の問題点

(トロン協会による調査, 1999年末～2000年頭, 日本)

# 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仕様カーネルの設計方針



## 設計コンセプト

- ▶ **弱い標準化**

## 設計方針

- ▶ ハードウェアの過度の仮想化を避け、ハードウェアに対する適応化を考慮する
  - ➔ **スケーラビリティの重視**
- ▶ アプリケーションに対する適応化を考慮する
- ▶ **ソフトウェア技術者の教育を重視する**
- ▶ 仕様のシリーズ化やレベル分けを行う
- ▶ 豊富な機能を提供する

# ITRON仕様カーネルの機能 (μITRON4.0仕様)



## カーネルの機能

- ▶ タスク管理機能
- ▶ タスク付属同期機能
- ▶ タスク例外処理機能
- ▶ 同期・通信機能
- ▶ 拡張同期・通信機能
- ▶ メモリプール機能
- ▶ 時間管理機能
- ▶ システム状態管理機能
- ▶ 割込み管理機能
- ▶ サービスコール管理機能
- ▶ システム構成管理機能

! I/O操作のための機能は規定されていない

## サービスコール数

- ▶ フルセット  
サービスコール : 166 (13 )  
静的API : 21
- ▶ スタンダードプロファイル  
サービスコール : 70 (13 )  
静的API : 11
- ▶ 自動車制御用プロファイル  
サービスコール : 43 (11 )  
静的API : 8
- ▶ 最小セット  
かっこ内は非タスクコンテキスト専用サービスコール (内数)

## ITRON仕様カーネルの開発状況



- ▶ 多くの ITRON仕様準拠製品
  - ▶ ITRON仕様準拠製品登録制度
    - 約40種類のプロセッサ用に約55の製品
  - ▶ その他にもいくつかの製品



メジャーな組込みシステム用プロセッサのほとんどすべて用に実装されている

- ▶ 海外メーカーが関与する実装が増える
  - ▶ 海外のソフトウェアメーカーによる実装
  - ▶ 海外半導体メーカーとの共同/委託による実装
- ▶ 極めて多くの社内用の実装
- ▶ いくつかのフリーの実装

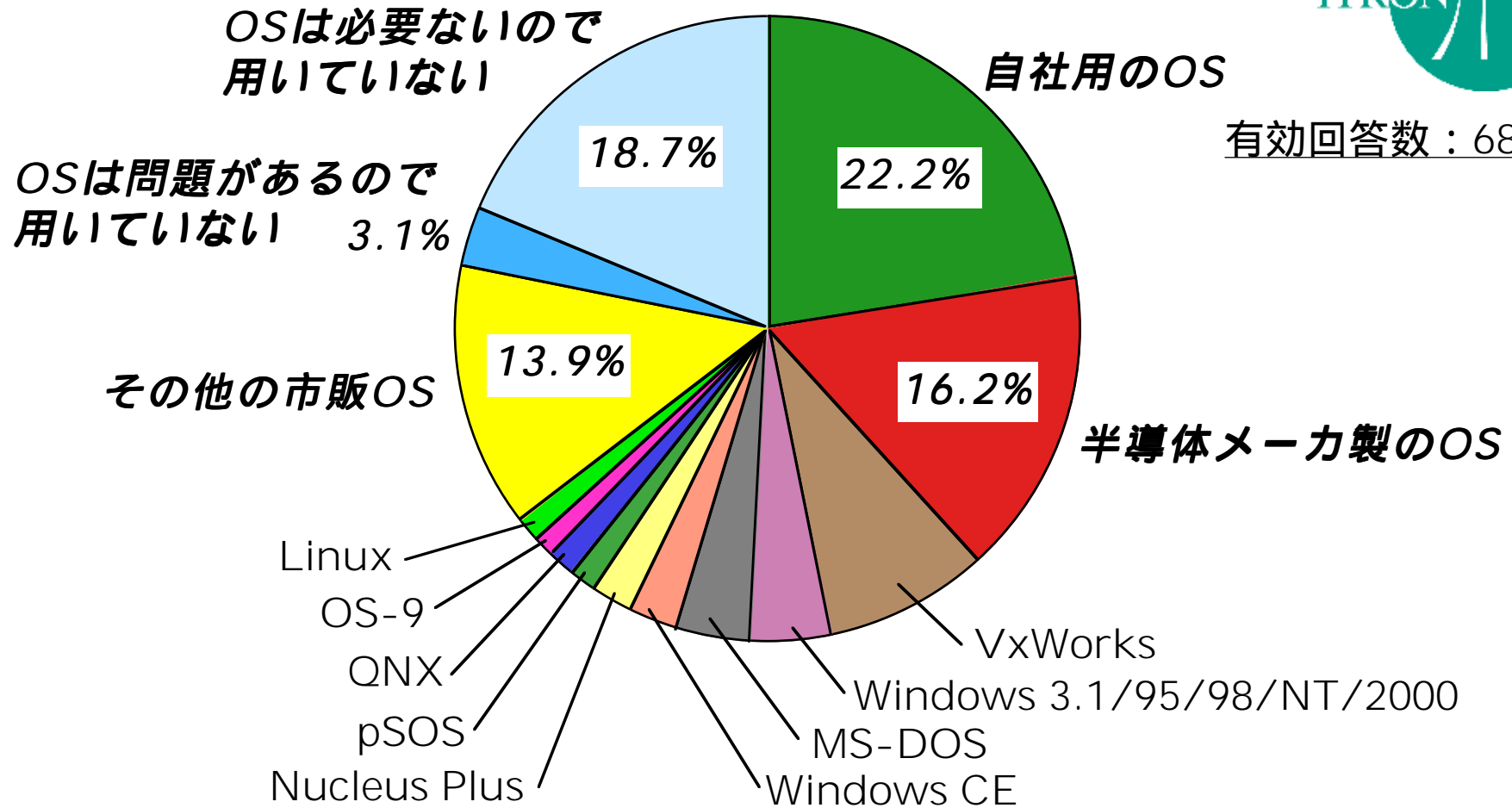
## ITRON仕様カーネルの利用状況



- ▶ 最も広く使われているリアルタイムOS仕様
- ▶ 特に，コンシューマ機器の分野において，ITRON仕様カーネルが使用されている比率が高い
- ▶ (社)トロン協会によるアンケート調査結果を紹介
  - ▶ 1999年11月～2000年1月に実施（毎年実施している）
  - ▶ 国内の組込みシステムの開発技術者を主な対象
  - ▶ 組込みシステム分野の展示会・セミナーでの配付とダイレクトメールを併用
  - ▶ 全回答者数は900名弱
  - ▶ 全データと分析結果はITRONホームページに掲載予定
  - ▶ 最も最近に開発した組込みシステムについて回答を求める（有効回答システム数は700弱）



有効回答数：684

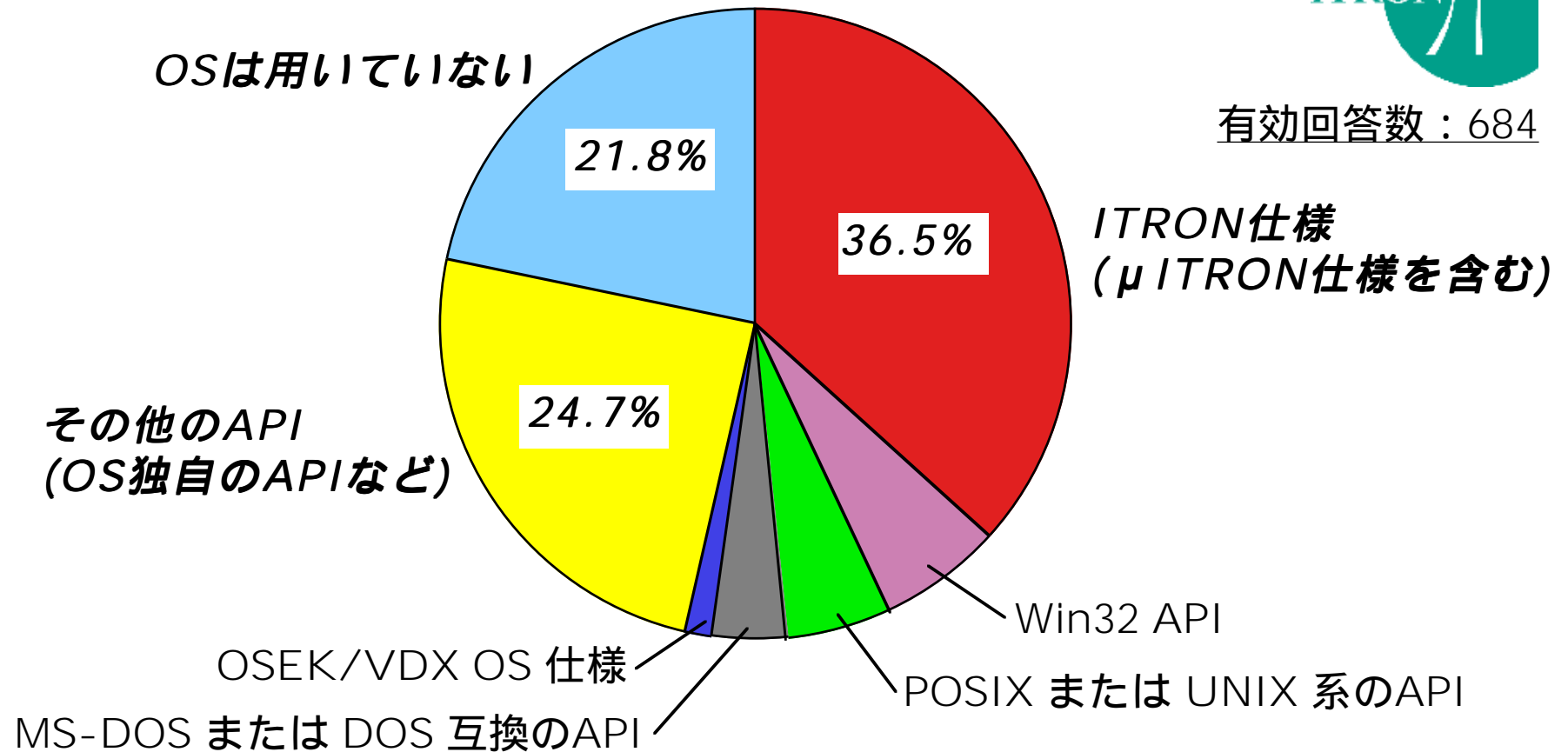


### 最近開発した組み込みシステムに組み込んだOS

(トロン協会による調査, 1999年末～2000年頭, 日本)

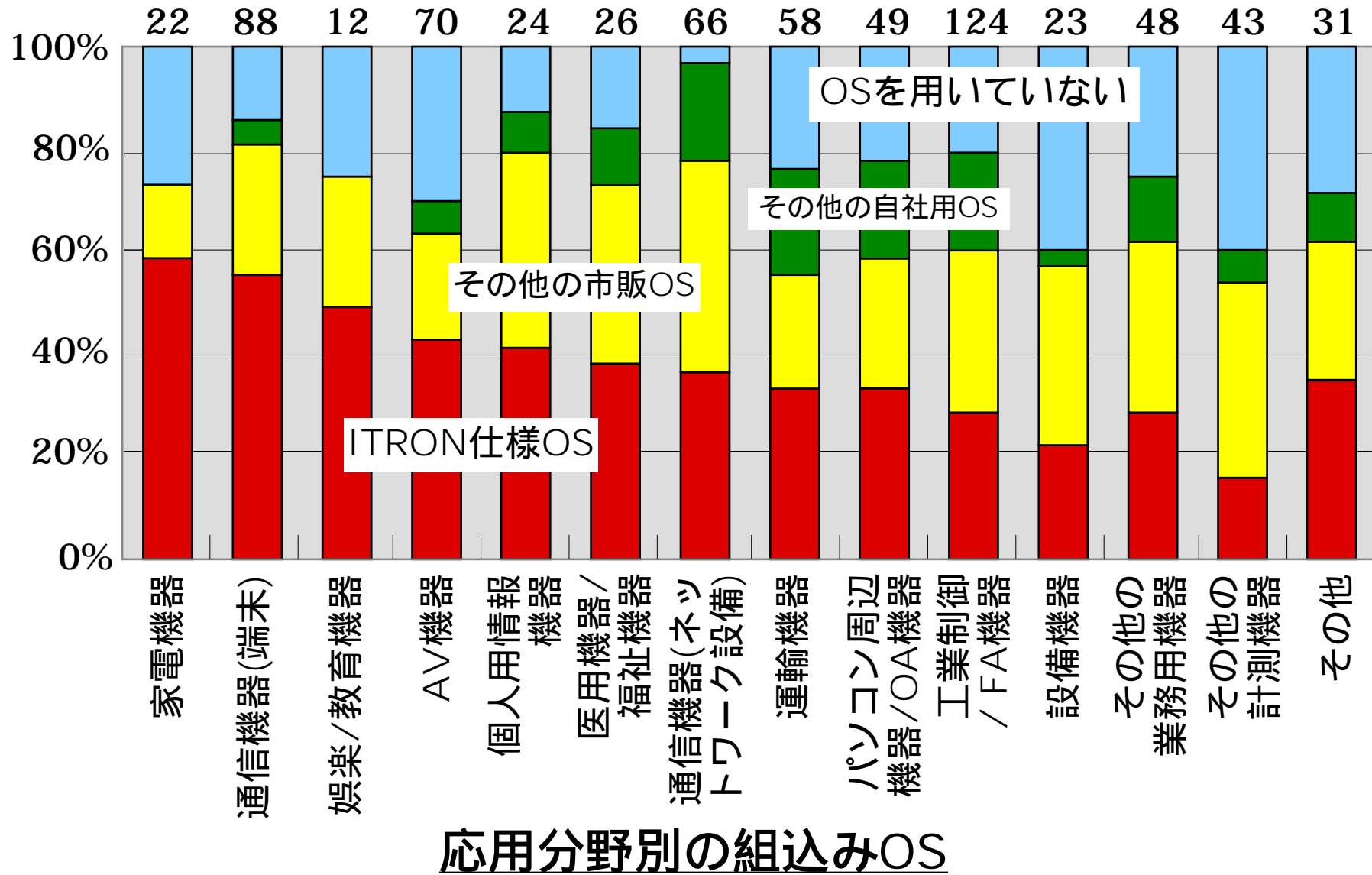


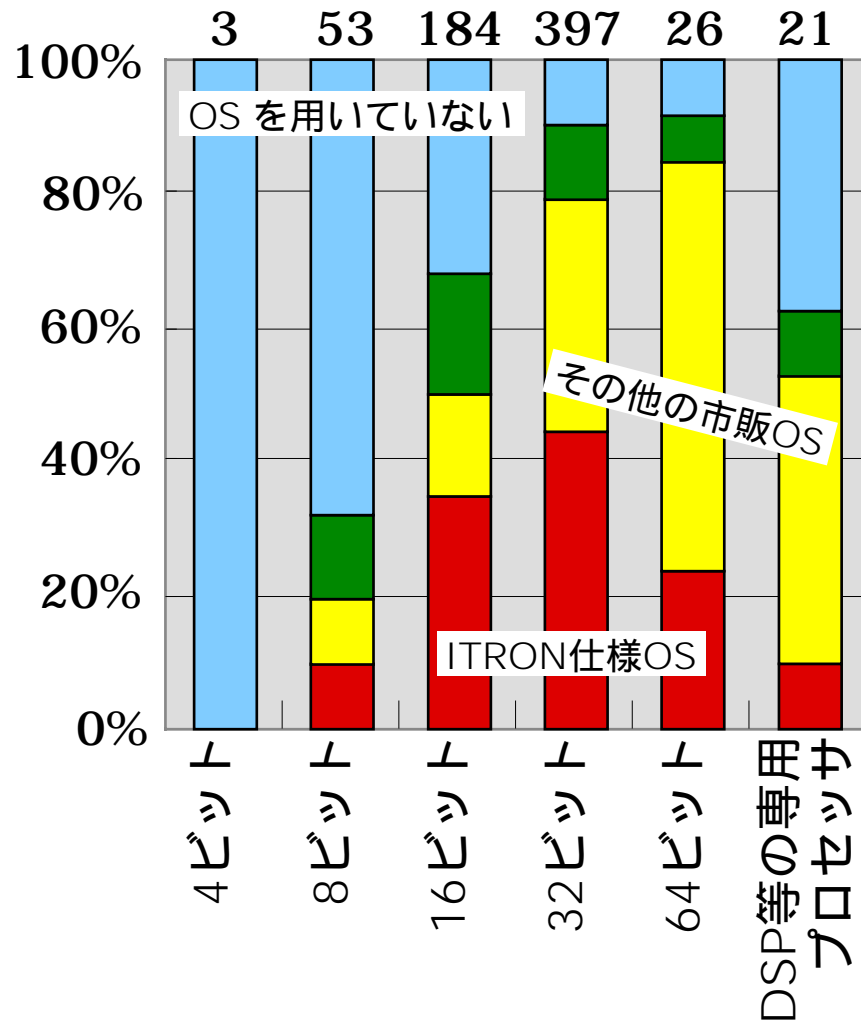
有効回答数：684



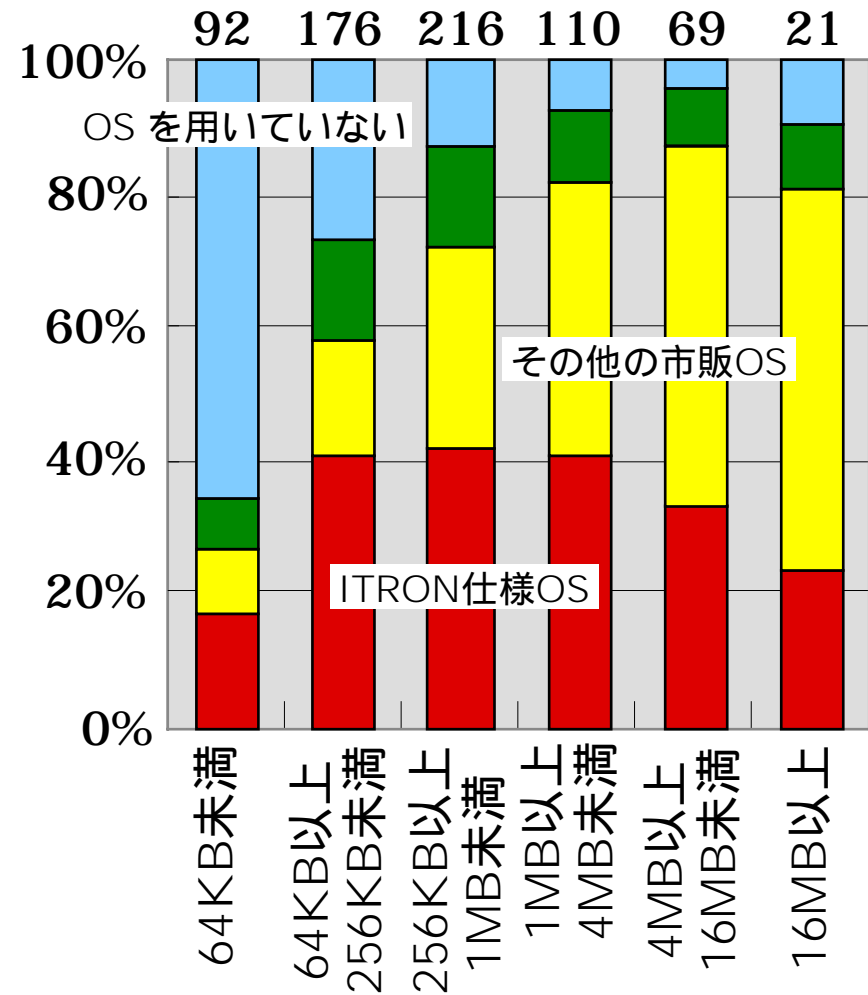
### 最近開発した組み込みシステムに組み込んだOSのAPI

(複数のAPIを持つ / 使用した場合には主に使用したもの)  
(トロン協会による調査, 1999年末 ~ 2000年頭, 日本)





CPUの規模別の使用OS



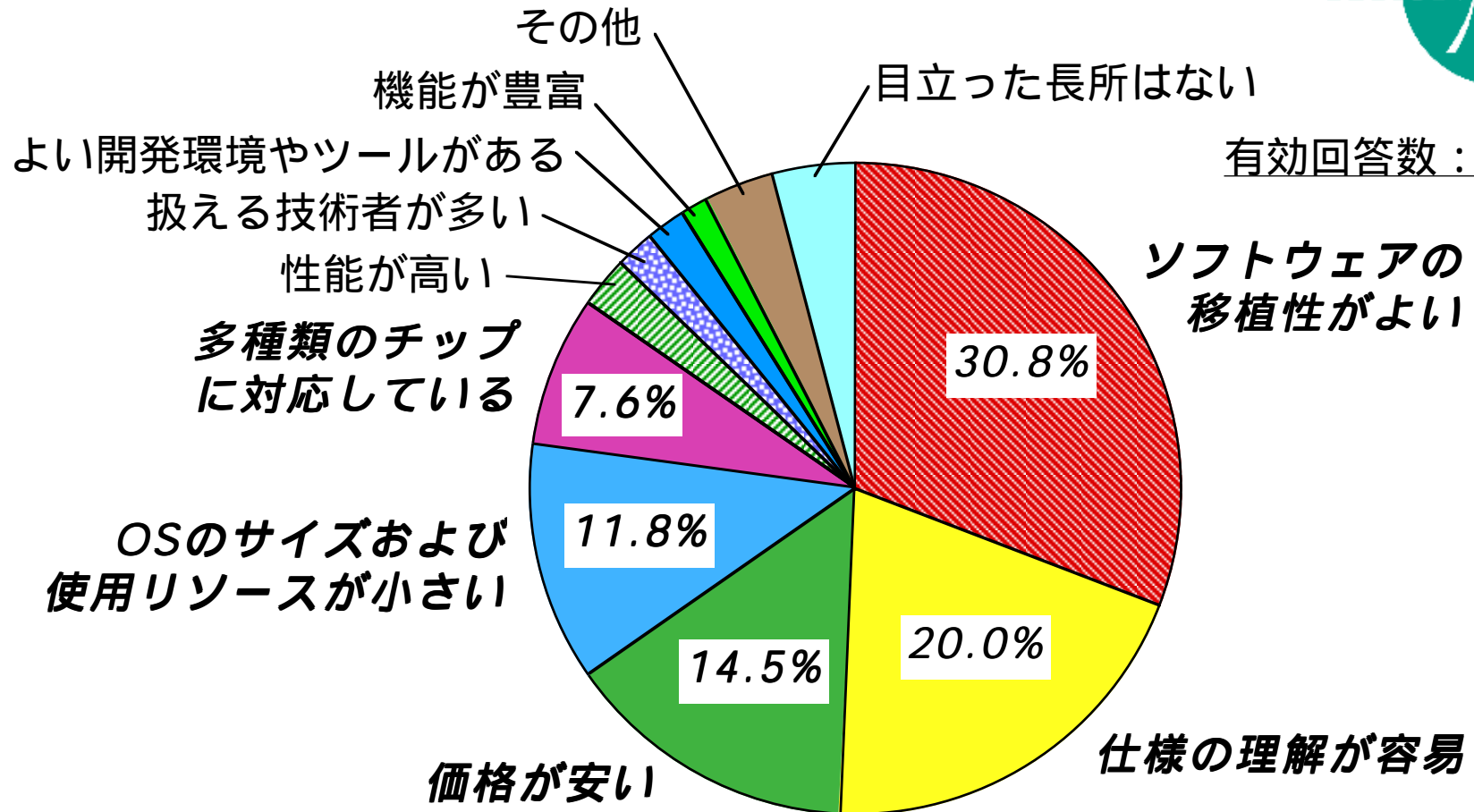
プログラムサイズ別の使用OS

■ は、その他の自社用OS

# ITRON仕様に対するユーザからの評価



有効回答数：638

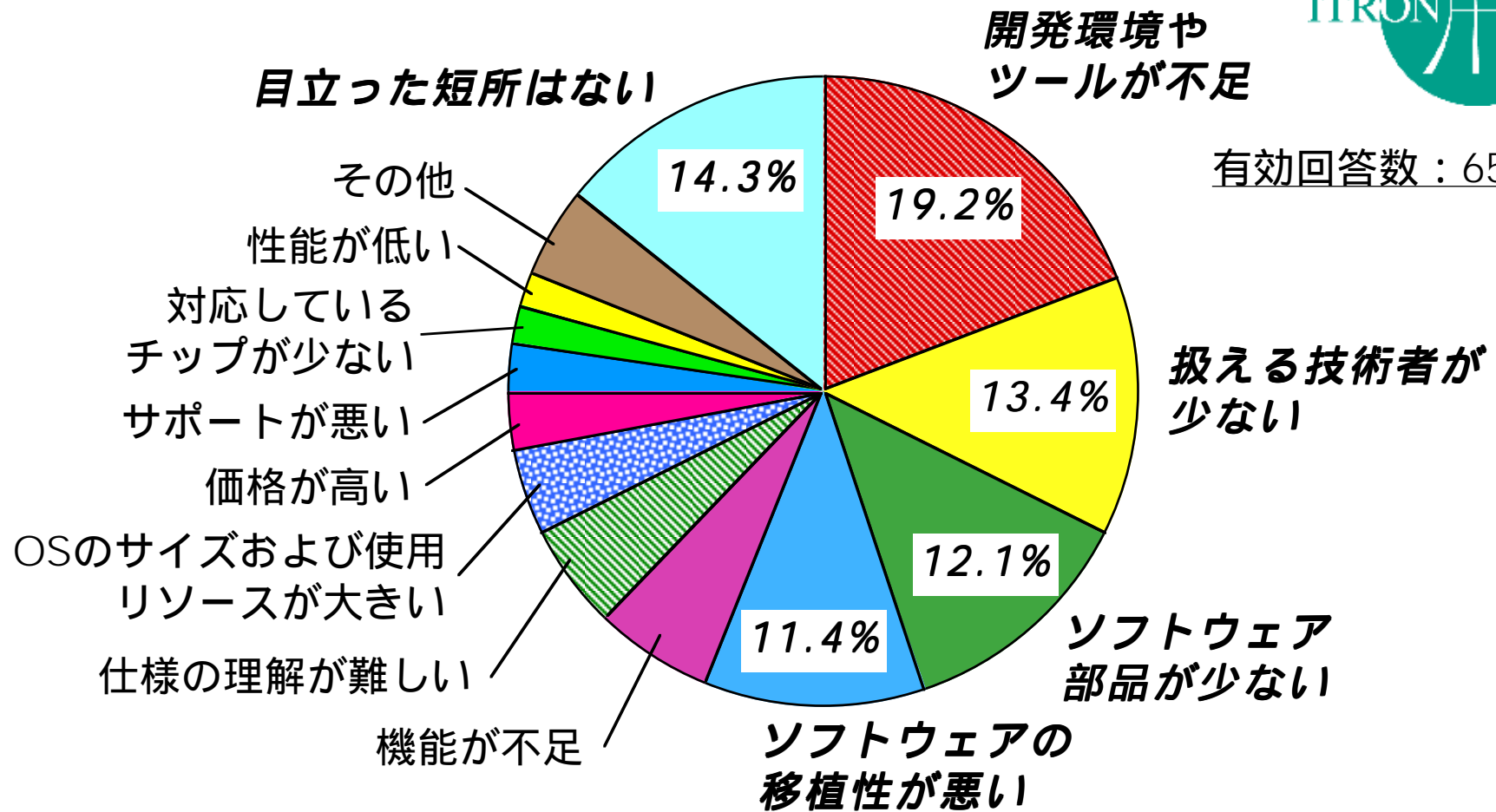


## ITRON仕様OSの長所

(トロン協会による調査, 1999年末～2000年頭, 日本)



有効回答数：651



### 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

## ソフトウェア部品流通の前提条件の整備



### ソフトウェアの移植性の重視

- ! 「弱い標準化」により移植性が阻害されているという指摘
- ▶ スタンダードプロファイルの考え方を導入
  - ➔  $\mu$ ITRON4.0仕様 1999年6月公開
- ▶ 仕様に合致していることを確認するための検定制度
  - ➔  $\mu$ ITRON4.0仕様 検定制様書 近日公開

### リアルタイム性の保証

- ? どのようにしてソフトウェア部品とアプリケーションの持つリアルタイム制約が両立することを保証するか?
  - リアルタイム制約を持ったソフトウェア部品の例：  
ソフトウェアモデム, 音声圧縮/解凍, MPEG
- ▶ ハードリアルタイム性を持ったシステムの構築手法
  - ➔ アプリケーション設計ガイドライン

## 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

## ソフトウェアの移植性の向上



- ▶ 基本的には標準化の度合いを強くすればよい



**一般には両立は困難**

- ▶ 適応化の利点（弱い標準化によって得られる）
  - ▶ 広範なスケラビリティの実現
  - ▶ ハードウェア（プロセッサ）の特徴を活かす
  - ▶ アプリケーション毎の要求への対応

### 両立可能な部分は両立させる

- ▶ 不必要な実装依存性の削減
  - ▶ 決めても差し支えない（＝適応化上問題が小さい）点については決める
- ▶ 使わない機能が容易に取り外せるAPI仕様に

# スタンダードプロファイルの導入



## コンセプト

- ▶  $\mu$ ITRONの適用分野の中で比較的大規模なシステムを想定し、標準的な機能セットを「強く」標準化
- ▶ 性能重視の小規模なシステムにはサブセットで対応
- ▶ 標準を越える要求のために拡張機能も定義

$\mu$ ITRON4.0仕様全体 ... 弱い標準化  
スタンダードプロファイル ... 強い標準化

## 言い換えると、

- ▶ 移植性を重視するソフトウェア (例: ソフトウェア部品) はスタンダードプロファイルの機能のみを用いる
- ▶ ソフトウェアの移植性を重視する分野向けのカーネルはスタンダードプロファイルに準拠して実装する



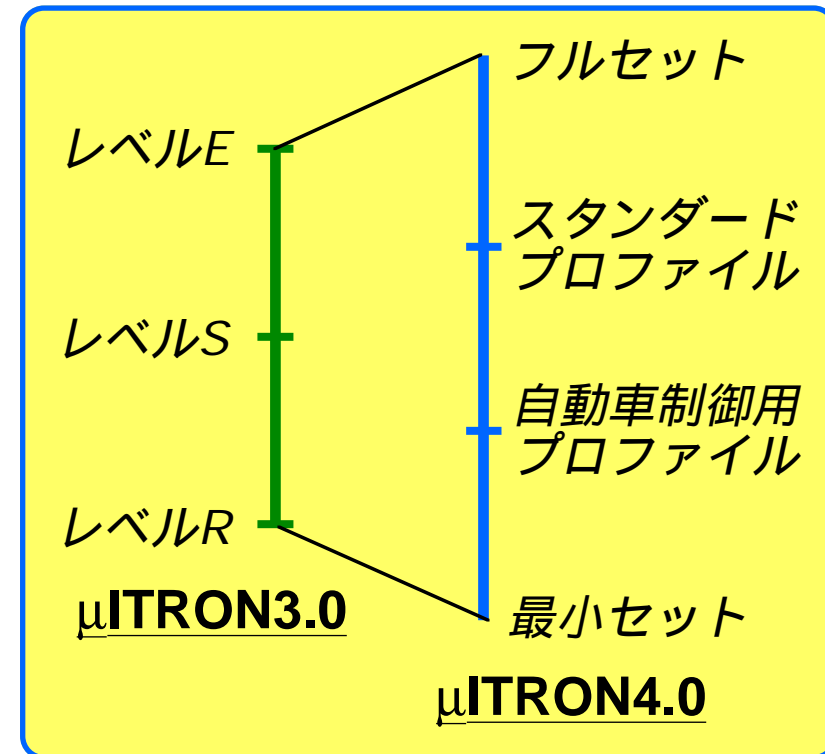
## より広いスケラビリティの実現

### μITRON3.0仕様よりも高機能化

- ▶ データキュー
- ▶ タスク例外処理機能
- ▶ システム状態参照機能
- ▶ 割込みサービスルーチン
- ▶ ハードリアルタイム対応
- ▶ ID番号の自動割付け

### 自動車制御用プロファイル

- ▶ 小規模なシステム向けの  
プロファイル規定



### より小規模なシステムへも適用可能に

- ▶ 待ち状態をオプションにして，休止状態を必須に

## μITRON4.0仕様に関する最近の状況



### 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（グレースシステム / ATI）

### μITRON4.0仕様準拠のフリーソフトウェア

- ▶ TOPPERS/JSPカーネル（豊橋技術科学大学）  
<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)

## ソフトウェア部品のインタフェースの標準化



! ソフトウェア部品の種類毎に標準化を行う必要性

→ 重要と考えられる分野から着手

▶ TCP/IPプロトコルスタックのAPI

→ ITRON TCP/IP API仕様 1998年5月に公開

▶ Java 実行環境とのインタフェースの標準化

→ JTRON2.0仕様 1998年10月に公開

→ 2.1仕様 2000年11月に公開

▶ デバイスドライバの標準化

... 標準化のニーズは高いが、デバイス毎の違いが大き  
く、性能を落さずに標準化する事は困難

→ デバイスドライバ設計ガイドライン 中間報告を公開中

▶ 組み込みシステム向けGUIパッケージのAPI

→ TRON GUI仕様 検討中

## ITRON TCP/IP API仕様



### TCP/IPプロトコルスタックのAPI

- ▶ ソケットインタフェースが主流だが，組込みシステム（特に小規模なもの）には不向きとの指摘
  - ▶ プロトコルスタック内で動的なメモリ管理が必須
  - ▶ RTOSのタスクモデルとUNIXプロセスモデルの違い

↓ 標準化の必要性

### ITRON TCP/IP API仕様

### ITRON TCP/IP API の設計方針

- ▶ ソケットインタフェースをベースに，ライブラリを載せてソケットインタフェースを実現できるものに
- ▶ 静的設定を活用
- ▶ ITRONと整合させるが，他のRTOSにも載るよう配慮

## JTRON仕様



### Java とリアルタイムOS の融合

- ▶ RTOS上に Javaの実行系を実現
- ▶ RTOSと Java の双方の利点を活用

リアルタイム性の必要な処理 ... RTOS上で実現  
ダウンロードするソフトウェア, GUI ... Javaで実現



*RTOS上のプログラムと Java のプログラムの通信方法は？*



標準化の必要性

### JTRON仕様

- ▶ ITRON上のプログラムと Java のプログラムの間の通信インタフェースを標準化
- ▶ JavaのスレッドはITRONのタスクに1対1に対応

# デバイスドライバ設計ガイドライン



## 現状の問題点

- ▶ ITRON仕様OSの短所として、デバイスドライバが揃っていないこと（流通していないこと？）が指摘されることが多い
- ▶ デバイスドライバは小規模なソフトウェアであるにもかかわらず、作成するのが難しいと言われている

## なぜ流通していないのか？

- ▶ いくつかの要因が複合？
- ▶ ITRON仕様OS用のデバイスドライバの標準的なモデルが示されていないことが原因の一つ
- ▶ 組み込みシステム用のデバイスドライバは、プラットフォーム（プロセッサ、OS、ボードなど）の多様性から、そのままの形が移植できることは稀で、流通性を阻害？

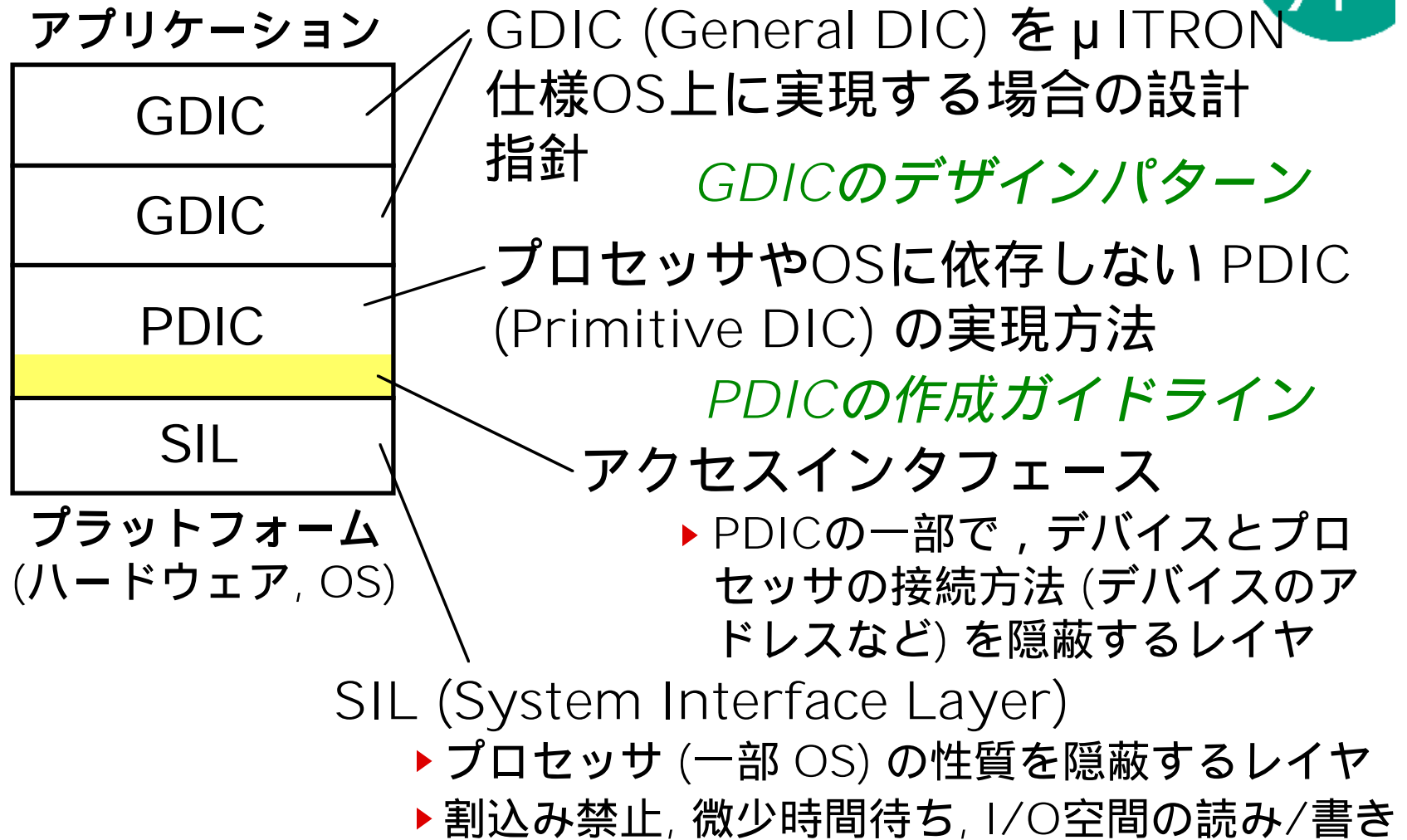


## ガイドライン策定の目的

- (a) ITRON仕様カーネル用のデバイスを扱うソフトウェアモジュールの標準的なモデルを提示
  - Device Interface Component (DIC) と呼ぶ
- (b) デバイスを扱う最低限のインタフェースソフトウェアを、プロセッサやOSに依存せずに記述する枠組みを与える (OSを使わない場合にも適用可能であるべき)
  - Primitive DIC (PDIC) と呼ぶ
  - ! PDICはデバイスメーカーが提供すべき
- (c) 熟練していない技術者にも安定した品質のDIC が開発できるよう指針を提示
- (d) デバイスを扱うソフトウェアに関連する用語の意味を標準化
  - ! APIの一律の標準化は意義が薄いと考えた



## DICの階層構造とガイドラインの内容



## 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

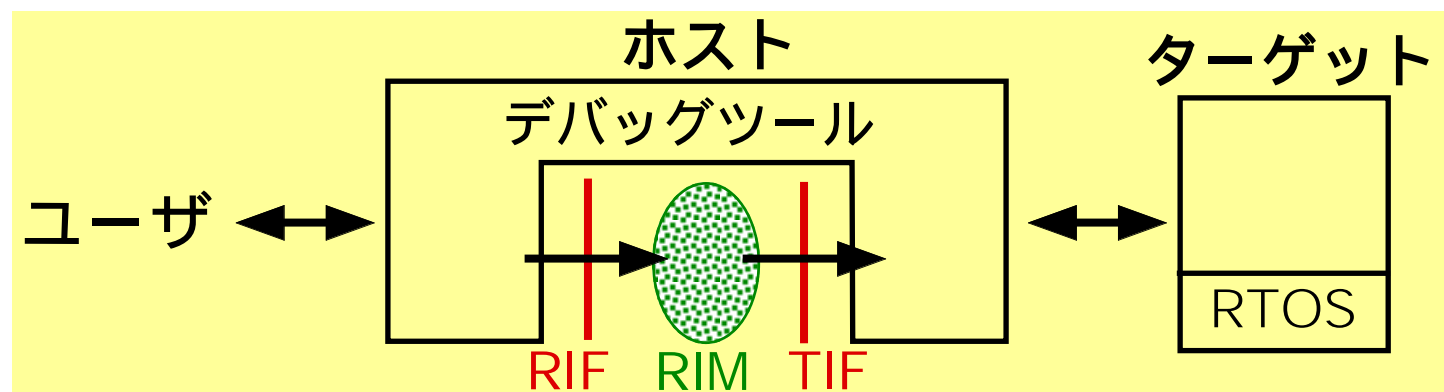
## 標準化の目標

- ▶ 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



## 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
- ▶ Windows DLL規定
- ▶ 動作履歴ファイル形式の規定

## 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](mailto: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 (?)

## 応用分野に特化した標準化



- ← 組み込みシステムの多様性
- ▶ 自動車制御応用
  - ▶ ITRONに限らずRTOSの適用が難しかった分野
  - ▶ RTOSの必要性が高まっていた
  - 自動車分野の技術者に呼びかけて、自動車制御用のリアルタイムカーネル仕様に対する要求を整理・仕様案を作成（1996年6月～1997年3月）
    - $\mu$ ITRON4.0仕様へ反映
  - ▶ 要求があれば，OSEK COMプロトコル（自動車内通信のプロトコル）のAPIを検討などを検討



## 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

- ▶  $\mu$ ITRON4.0仕様 1999年6月公開
- ▶  $\mu$ ITRON4.0仕様検定仕様書 近日公開予定

### 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 1998年10月 → 2.1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

### 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デバッグン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 検討中

### 応用分野に固有の要求への対応

- ▶ RTOS自動車応用技術委員会 完了 →  $\mu$ ITRON4.0仕様に

## ソフトウェア部品の整備



### JCGプロジェクト

- ▶  $\mu$ ITRON仕様カーネル上で動作するソフトウェ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年度

ITRON TCP/IP API仕様

JTRON仕様

要求事項

整合性

2001年度

次の展開

2000年度 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年度 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