

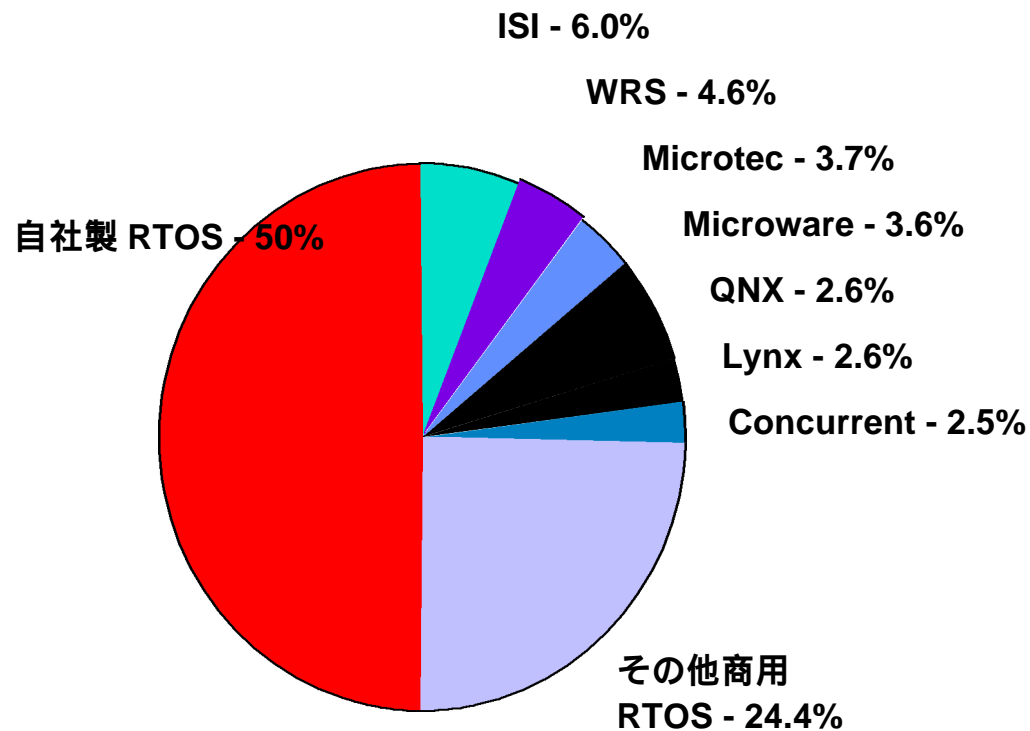
# eCos $\mu$ ITRON仕様互換APIと コンフィギュレーション機構



日本シグナスソリューションズ

ゼネラルマネージャ 伊藤紀彦

# eCos開発に至る背景(1)



RTOS市場が断片化

ソフトウェア資産の共通化が出来ず産業の発展を阻害

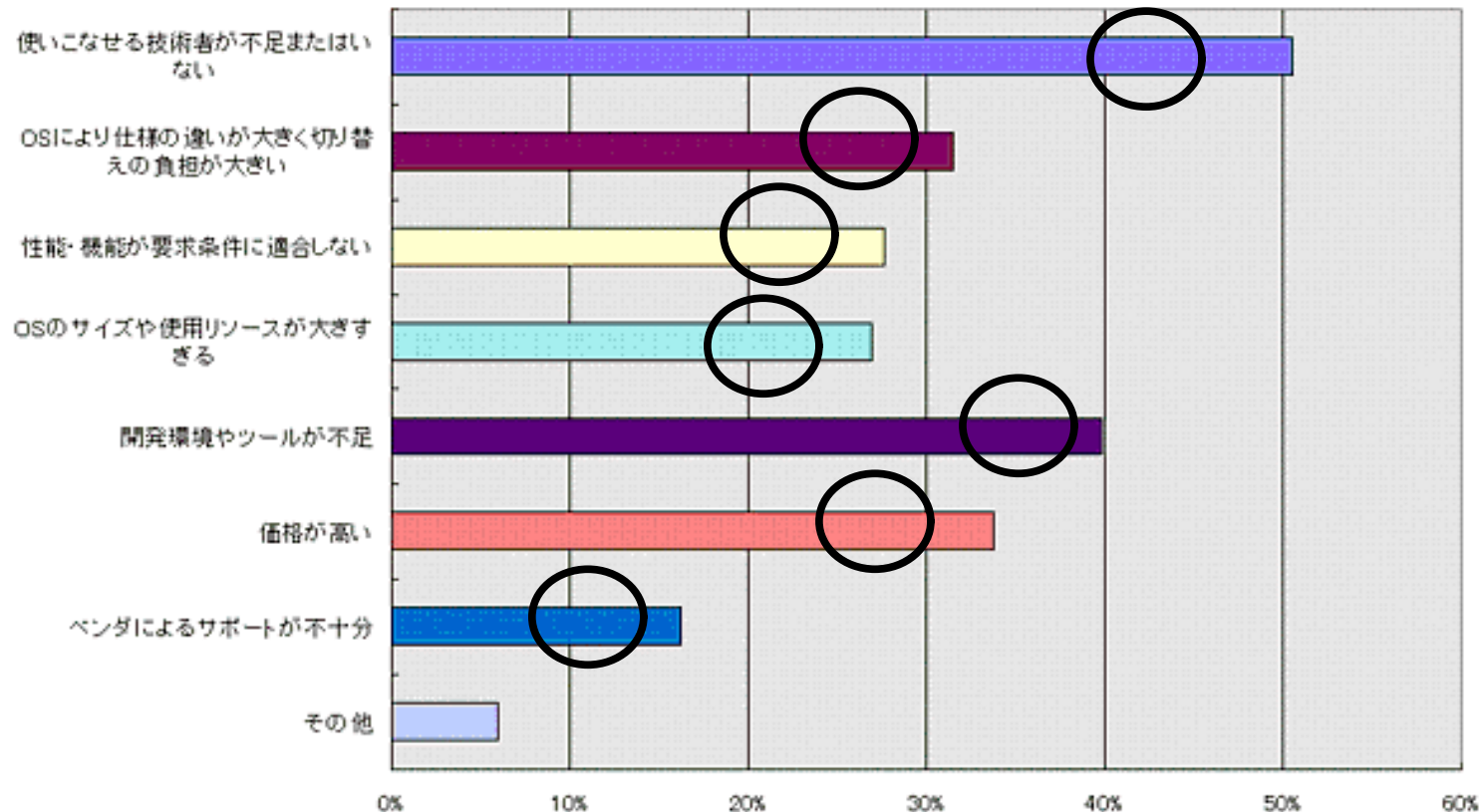
一つのRTOSでは多様なニーズに応えられない

出典:1997年Venture Development Corporation

# eCos開発に至る背景(3)

リアルタイムOSの問題点 (複数回答)

有効回答数: 263



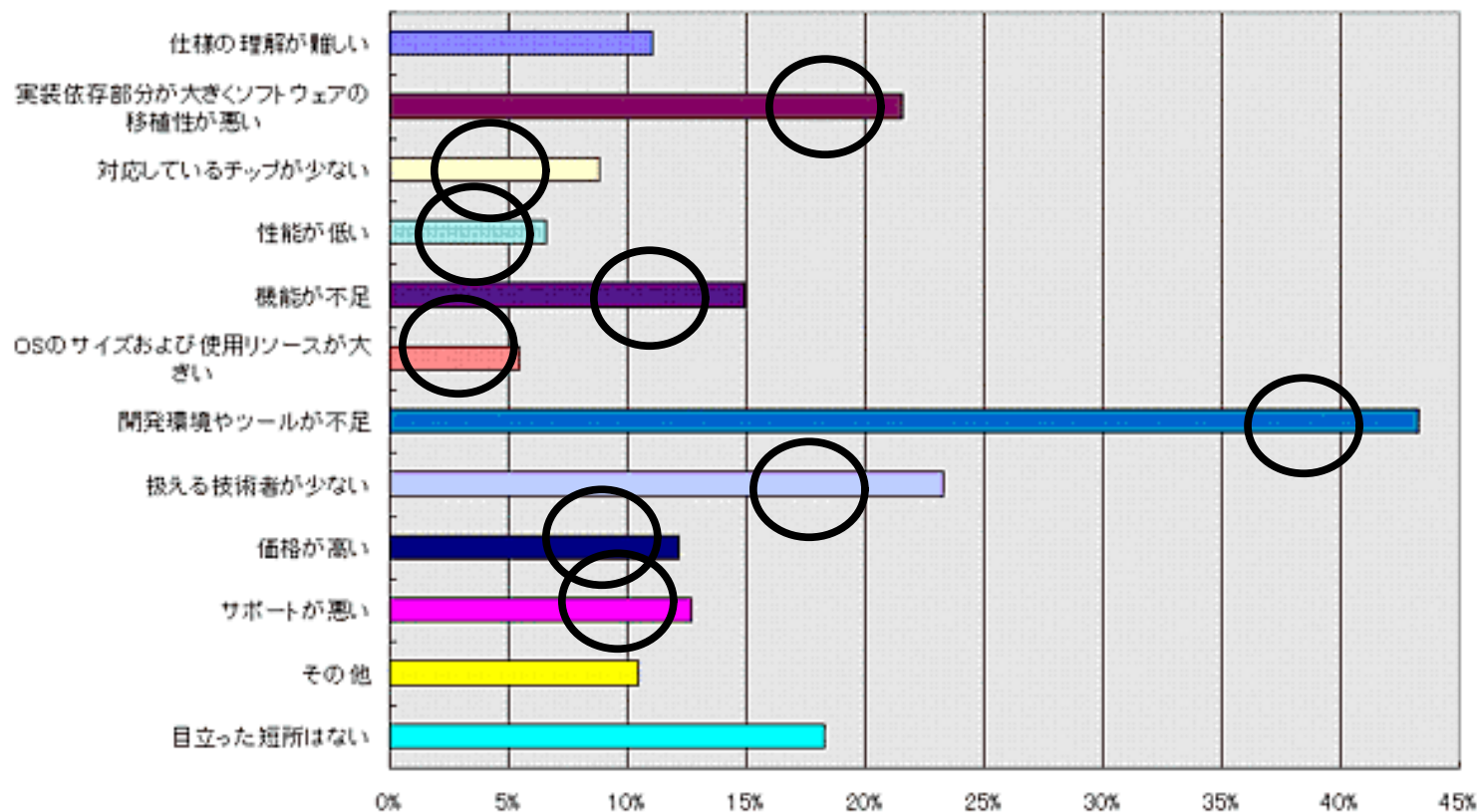
出典: <http://www.ertl.ics.tut.ac.jp/ITRON/survey97>

○ eCosで対策

# eCos開発に至る背景(4)

ITRON仕様OSの短所(複数回答)

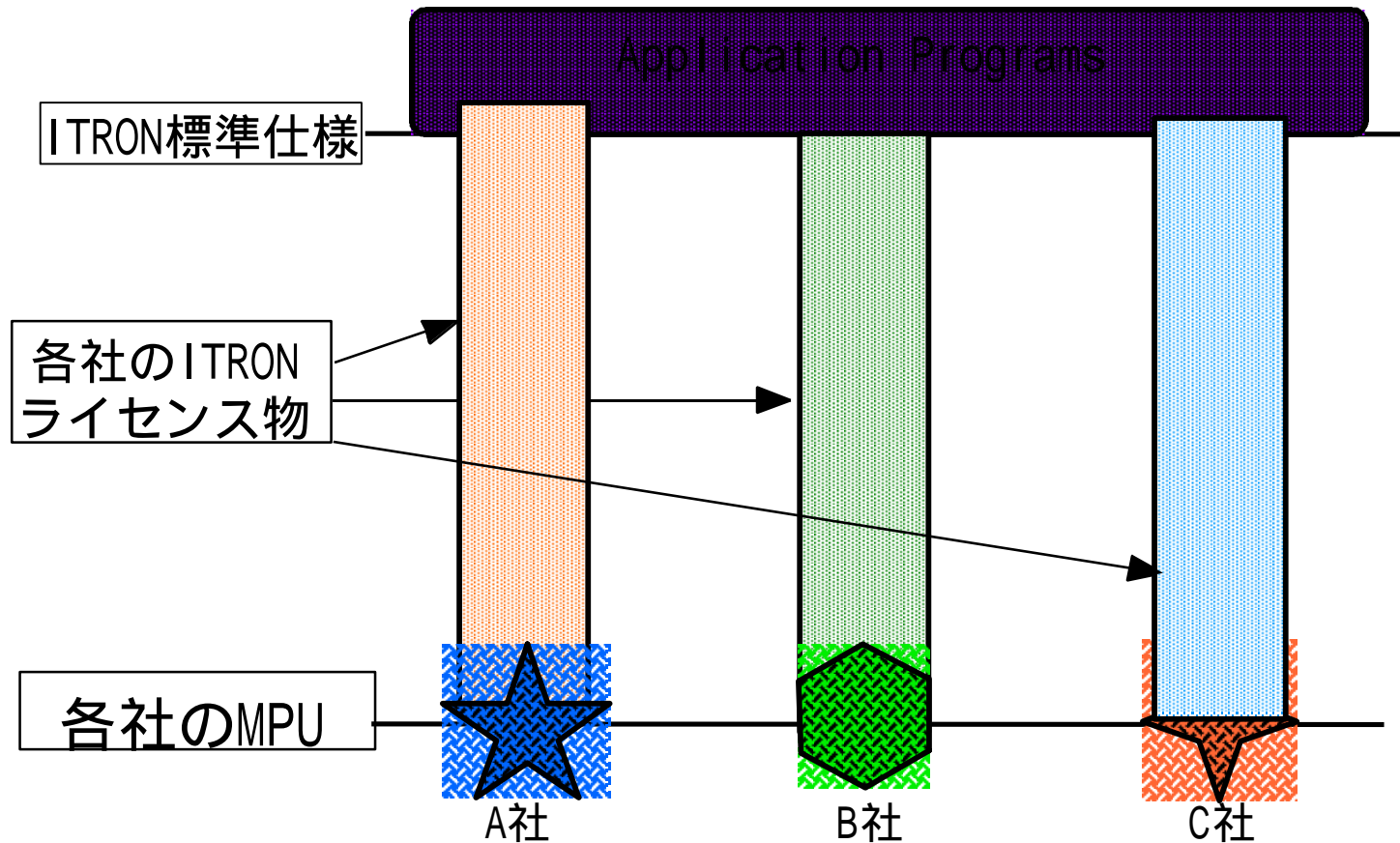
有効回答数: 180



出典: <http://www.ertl.ics.tut.ac.jp/ITRON/survey97>

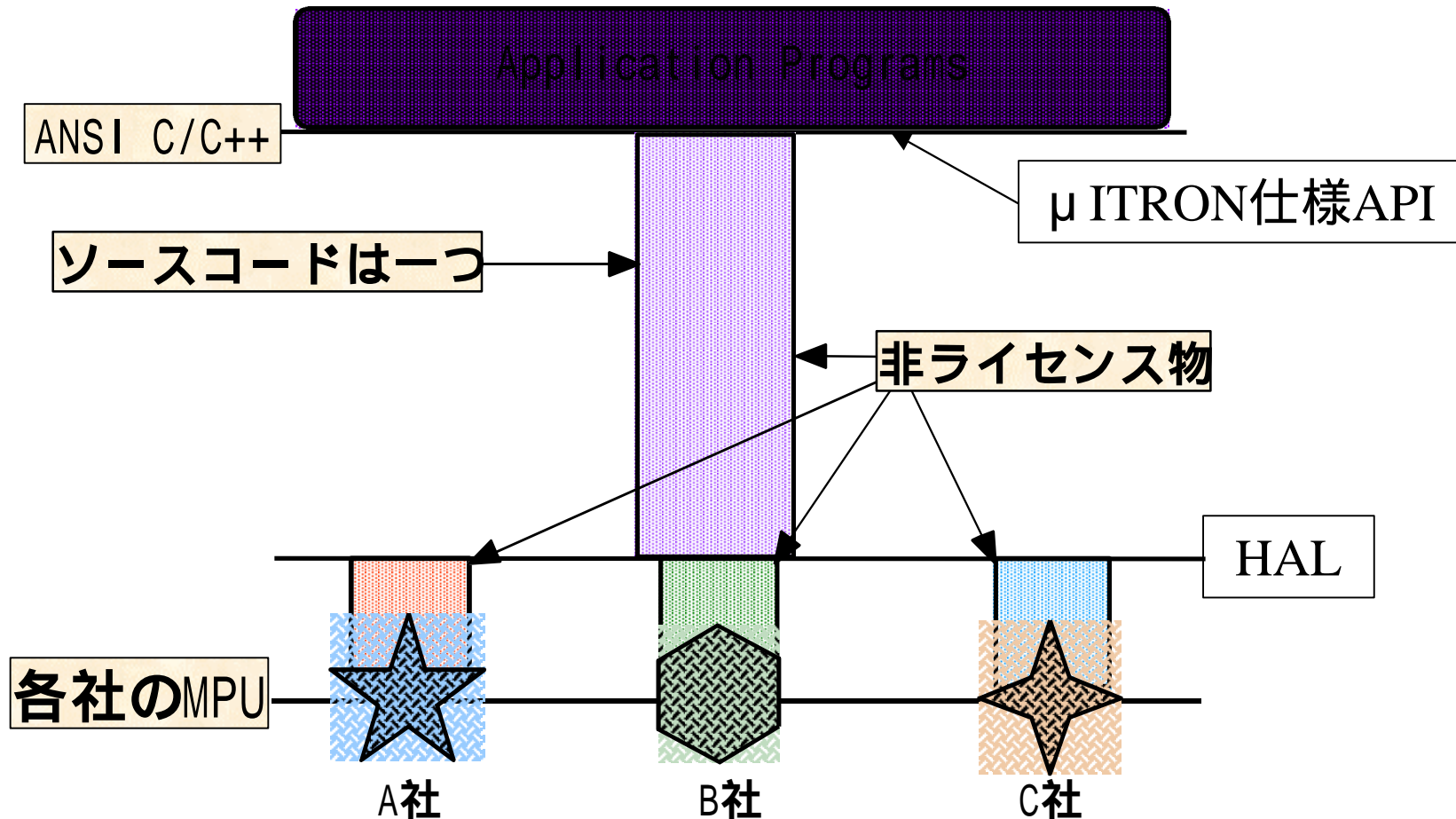
○ eCosで対策

# eCos開発に至る背景(5)



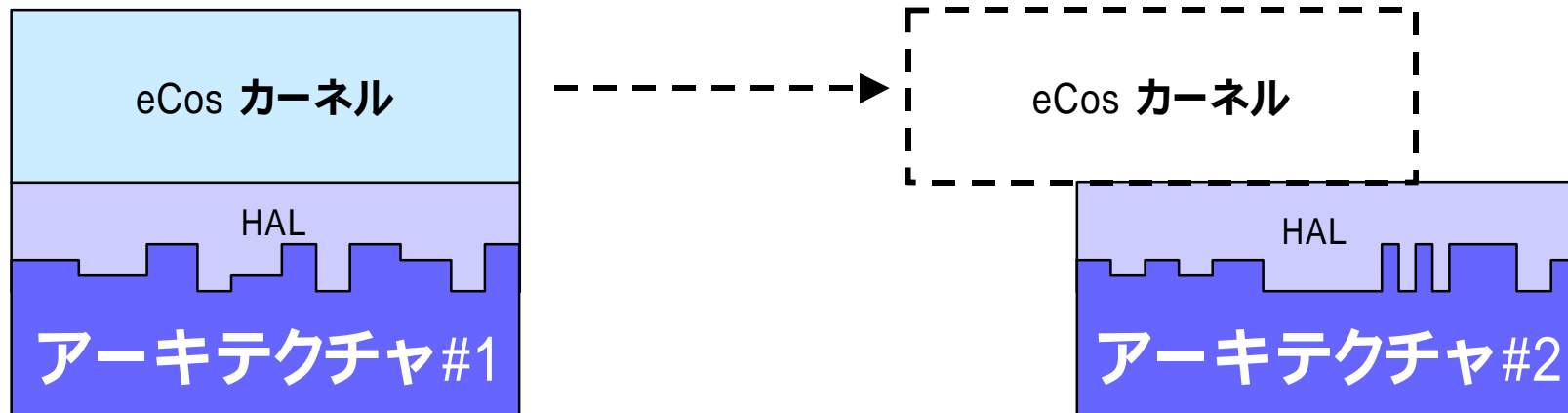
出典: 97年ITRONオープンセミナー(日本シグナス)

# eCos開発に至る背景(6)



出典: 97年ITRONオープンセミナー(日本シグナス)

# ハードウェア抽象化レイヤ(HAL)



HALの用法:

全てのハードウェアに依存するコードをシステムティックかつ高品質でポーティングする。

HALの解説:

アーキテクチャ固有部分は全体のわずか**5%程度**。

標準ハンドラとしては、タイマ、状態操作、割込、ボード及びMCUレベルの初期化、例外処理及びスレッド操作がある。

# eCos開発に至る背景(7)

世の中のトレンドを見る

インフラは無料になる — C、RTOS、TCP/IP、  
道路、インターネット、究極は空気や水と同じ

サービスの価値が増大する

ニーズはApplication Specificにシフトする

Time to Marketは永遠の課題

Open Source技術革新、その速度に注目

Internet Powerの活用



# 標準化アプローチの違い

## 1. 1社の力で支える事実上の標準

Windows, MS Office, Postscript

## 2. 多数の企業で推進する標準化委員会

ANSI C/C++仕様標準化委員会

トロン協会ITRON専門委員会

## 3. Open Source Model を背景に

ソースコードの公開

技術革新の対流促進

ロイヤリティフリー

すぐ使えるソフトをフリーダウンロード

多数のマイコンターゲット



# 問題解決の具体的方策(1)

## 開発環境やツールが不足

- 世界初のGUI-RTOSコンフィギュレータ
- タスクウェアGUIデバッガ

## 扱える技術者が少ない

- フリーソフト化することで世界中に技術提供

## ソフトの移植性が悪い

- HALの導入、C++の活用で移植性飛躍的に向上

## 機能が不足

- Configurableにする事でApplication SpecificなRTOSを構成
- Middleware 提供メーカーのリクルート

## 問題解決の具体的方策(2)

### サポートが悪い

- シグナスからMission Critical Support提供、1~5 day response
- インターネットを活用した先進的サポートインフラストラクチャ

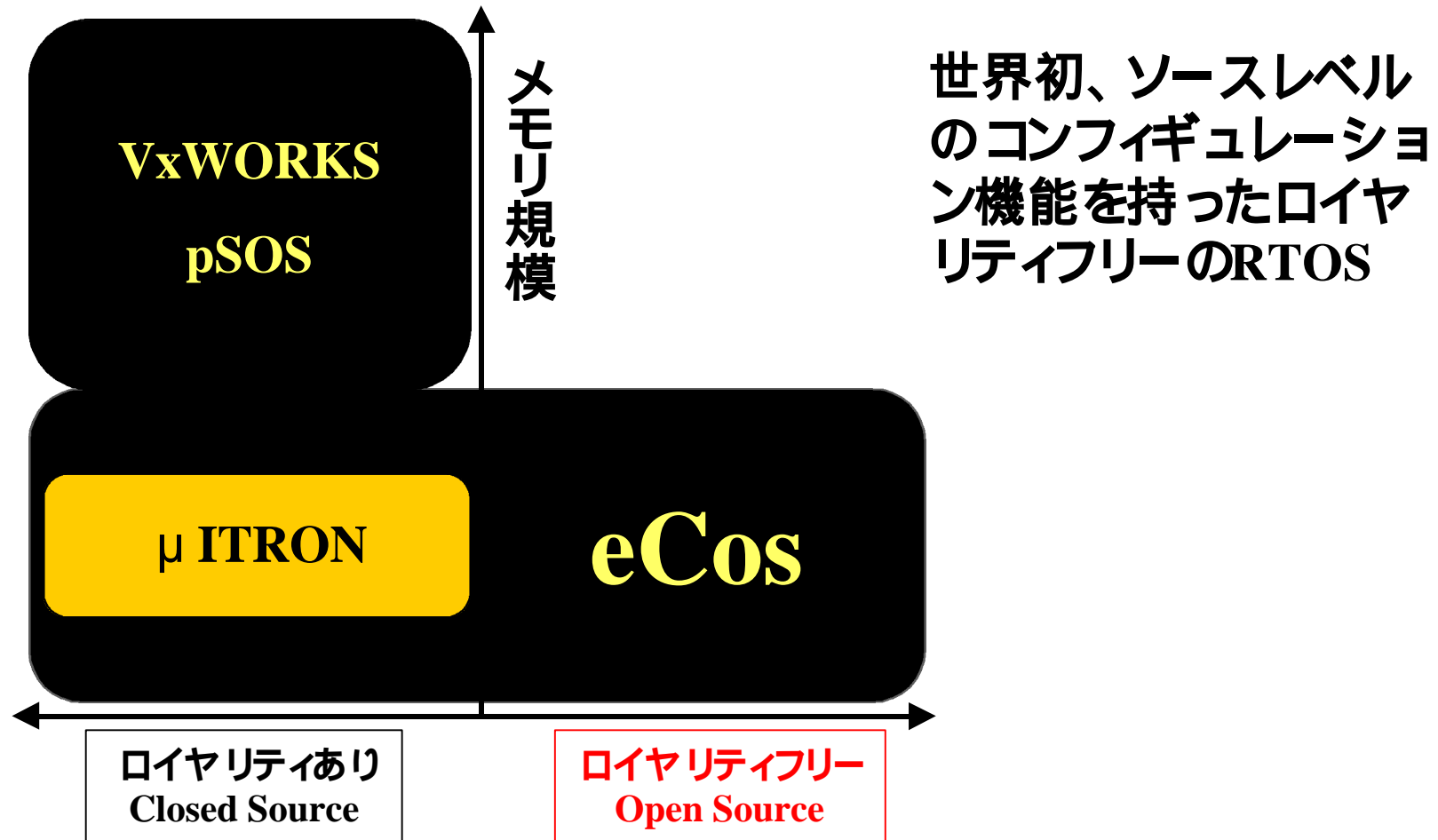
### 価格が高い

- C/C++コンパイラ、ランタイムソフトとも**ロイヤリティフリー**

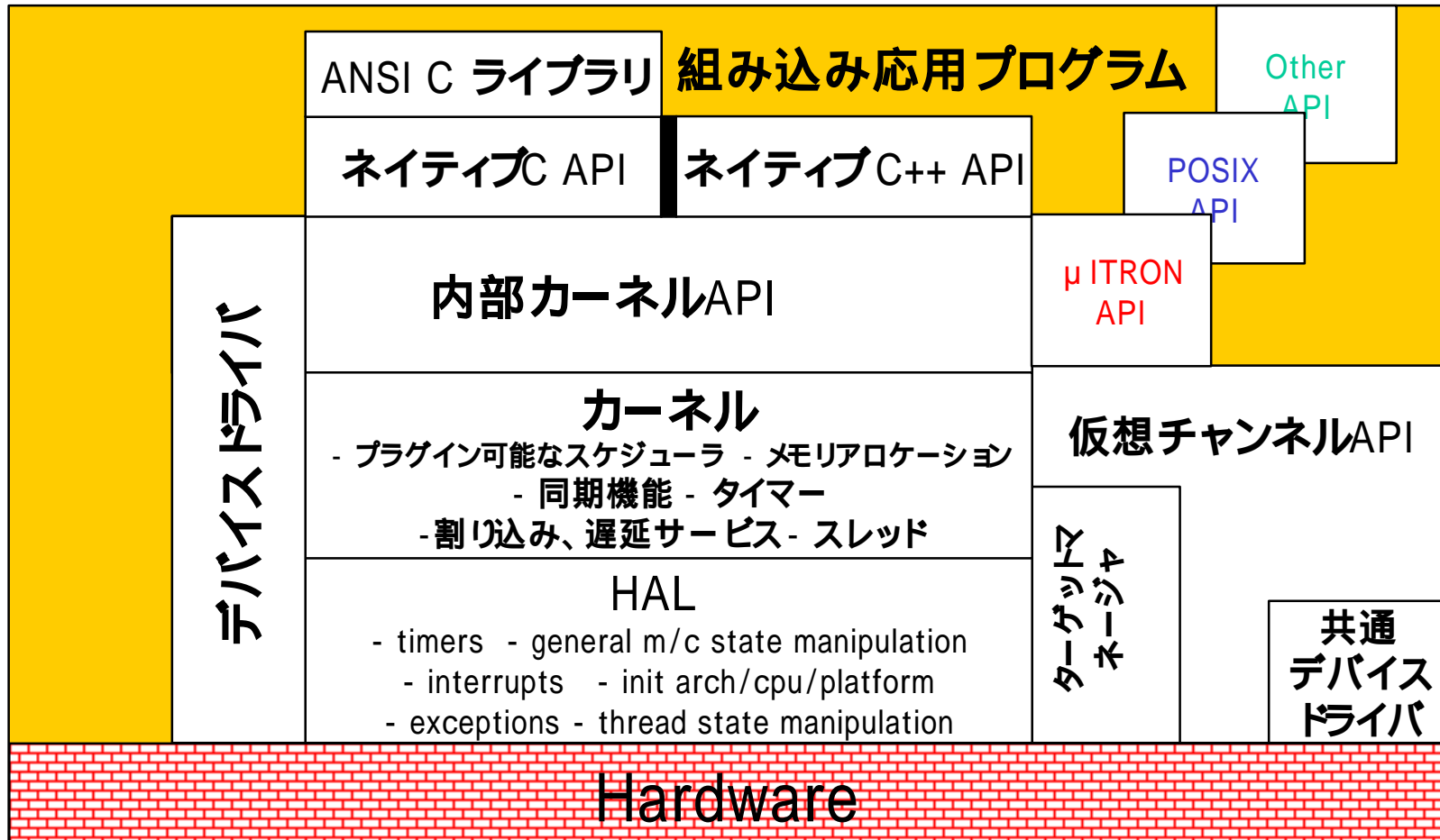
### 対応しているチップが少ない

- 今日対応:**TX39、MN10300、Power PC、ARM7、SPARClite、VR4300、SH-3、他**
- 他のマイコンへの**移植期間2~3ヶ月**
- **移植費:\$75,000 ~ \$100,000**

# eCosの位置づけ

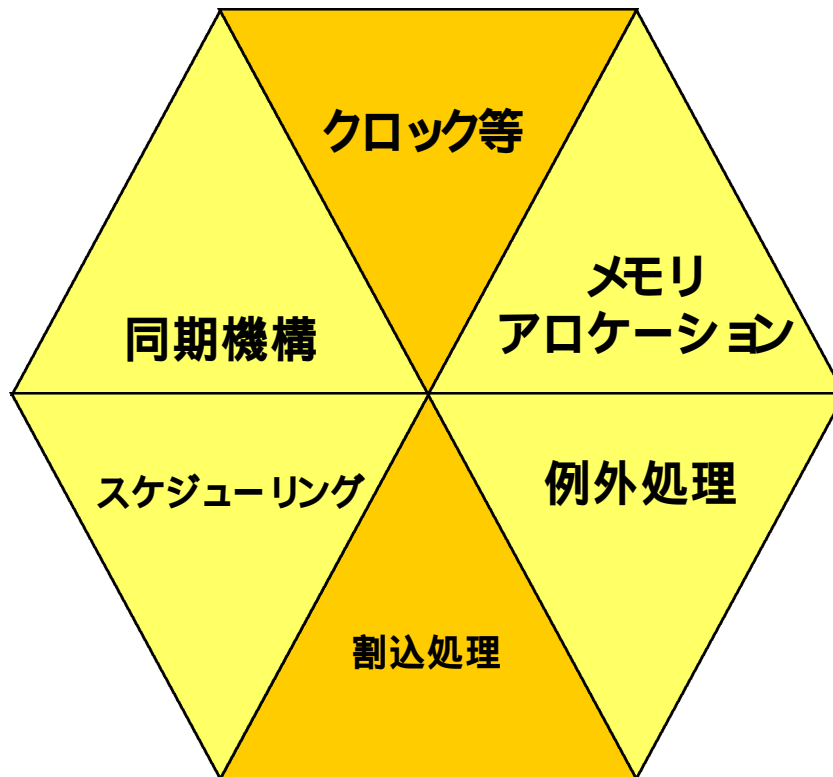


# eCos Kernel全体像



# カーネルの概要

## 主なカーネルの要素



プリエンプティブ、マルチスレッド、プライオリティレベル制御

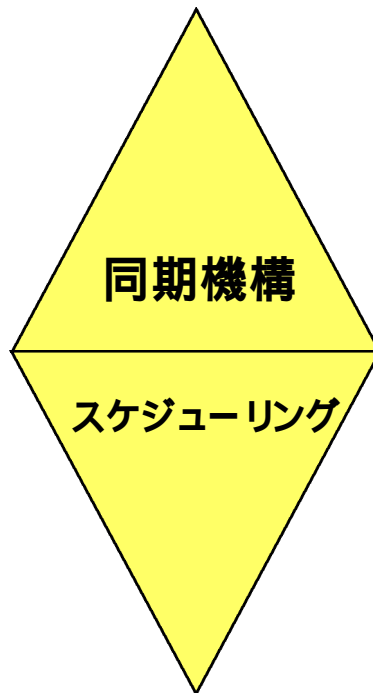
ソースレベルコンフィギュレーション

様々なAPIに対応可能

その一つが  $\mu$ ITRON仕様API

Application Specific OS

# 同期とスケジューリング

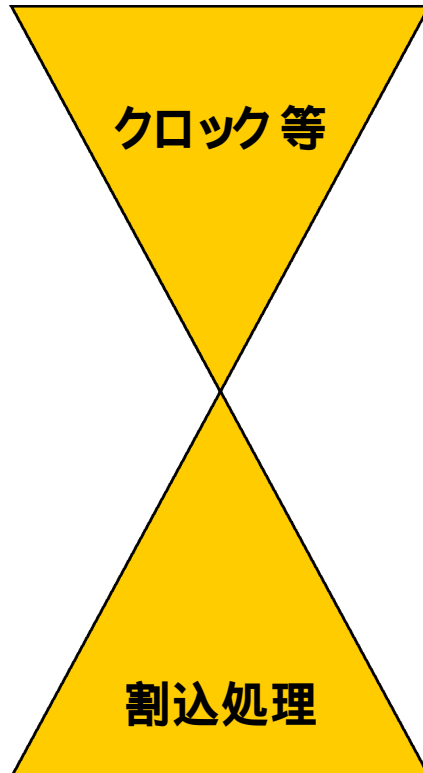


ミューテックス  
カウンティング、バイナリセマフォ  
条件変数  
メッセージボックス  
イベントフラグ  
プライオリティ継承

---

ビットマップスケジューラ  
マルチレベル・キュースケジューラ  
ロッタリスケジューラ(実験的)  
ユーザによるスケジューラの追加が可能  
スケジューラのロック  
Deferred service routine support

# クロックと割込



カウンタ

リアルタイムクロック

可変分解能クロック

カウンタからクロックへの変換をサポート

カウンタ、クロック、イベントとアラームとの接続

---

割込の有効、無効、リストアなどの操作

割込、ベクタサービス処理のサポート

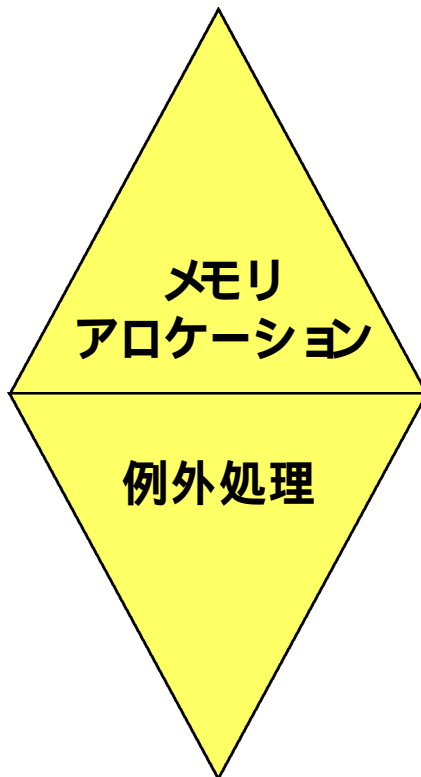
割込のマスク操作

割込のアクリッジ処理

割込のプライオリティ設定



# メモリ割り当てと例外処理



固定ブロック malloc

可変ブロック malloc

Free

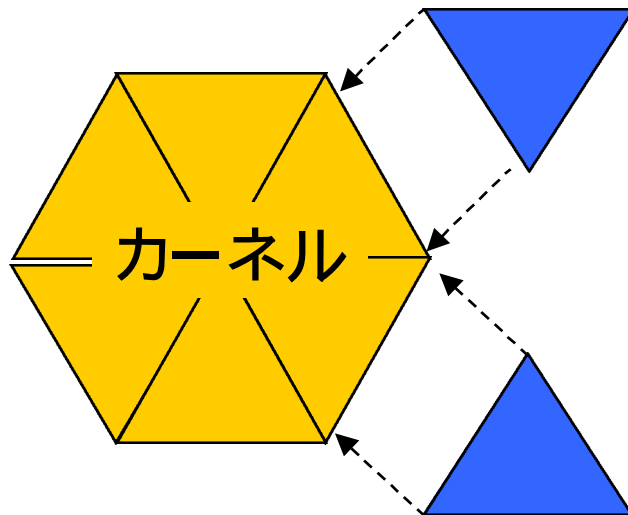
メモリプール

---

グローバルまたは、スレッドレベルの例外処理

# ミドルウェアコンポーネント

EBS社から提供



<http://www.etcbin.com>

シグナスでも再販提供

## インターネットスタック

TCP/IP  
FTP  
PPP  
UDP  
HTTP  
Web Server

## ファイルシステム コンポーネント

フラッシュファイルシステム  
DOSファイルシステム  
仮想ファイルシステム  
RAM ディスク  
CD ROM

# ミドルウェアコンポーネント

エーアイコーポレーションから提供

機器組み込み用TCP/IP

USNET

組み込み用WEBサーバ&インターネットアクセス

USNET IAP

ネットワークマネージメントプロトコル

USNET SNMP

PC互換ファイルシステム

USFiles

超高速浮動小数点ライブラリ

GOFAST

IrDA プロトコルスタック

JetBeam

フラッシュファイルシステム

FlashFX

RS-485ネットワークプロトコル

9-Bit Solution

# eCosのGUIコンフィギュレーション

The screenshot shows the eCos Configuration Tool interface. The left pane displays a tree view of configuration options, including 'eCos ulTRON Compatibility Layer' and 'Check Strict ulTRON Standards Conformance'. The right pane displays the 'eCos Reference Manual' for Chapter 6, 'μITRON API', with a table of contents and a table of properties for the selected macro.

**eCos Reference Manual**

[Prev](#) [Next](#)

## Chapter 6. μITRON API

**Table of Contents**

- [Task Management Functions](#)
- [Task-Dependent Synchronization Functions](#)
- [Synchronization and Communication Functions](#)
- [Extended Synchronization and Communication Functions](#)
- [Interrupt Management Functions](#)
- [Memory pool Management Functions](#)
- [Time Management Functions](#)
- [System Management Functions](#)

Property	Value
Type	Boolean
Value	False
Default Value	False
Macro	CYGIMP_ULTRON_STRICT_CONFORMANCE
File	C:\Program Files\Cygnus Solutions\ecos\packages\compat\uitron\v1_0\include\pkgonf\uitron.h
Defined at line	121
URL	ref/ecos-ref/micro-itron-api.html
requires	CYGVAR_KERNEL_COUNTERS_CLOCK
requires	CYGSEM_KERNEL_SCHED_MLQUEUE
precludes	CYGSEM_KERNEL_SCHED_TIMESLICE
requires	CYGFUN_KERNEL_THREADS_TIMER

# CDL (Component Description Language)

```
/*-----  
* {{CFG_DATA  
  
cdl_component CYGPKG_KERNEL_SCHED {  
  display "Kernel schedulers"  
  type dummy  
  parent CYGPKG_KERNEL  
  description "  
    The eCos kernel provides a choice of schedulers. In addition  
    there are a number of configuration options to control the  
    detailed behavior of these schedulers.  
  "  
  doc ref/ecos-ref/ecos-kernel-overview.html#THE-SCHEDULER  
}  
  
cdl_option CYGSEM_KERNEL_SCHED_MLQUEUE {  
  display "Multi-level queue scheduler"  
  type radio  
  parent CYGPKG_KERNEL_SCHED  
  description "  
    The multi-level queue scheduler supports multiple priority  
    levels and multiple threads at each priority level.  
    Preemption between priority levels is automatic. Timeslicing  
    within a given priority level is controlled by a separate  
    configuration option"  
  doc ref/ecos-ref/ecos-kernel-overview.html#THE-SCHEDULER  
}
```

```
cdl_option CYGNUM_KERNEL_SCHED_PRIORITIES {  
  display "Number of priority levels"  
  type count  
  legal_values 1 to 32  
  parent CYGPKG_KERNEL_SCHED  
  #active_if CYGINT_KERNEL_SCHED_PRIORITY_SCHEDULER  
  description "  
    This option controls the number of priority levels that are  
    available. For some types of scheduler including the bitmap  
    scheduler this may impose an upper bound on the number of  
    threads in the system. For other schedulers such as the  
    mlqueue scheduler the number of threads is independent from  
    the number of priority levels. Note that the lowest priority  
    level is normally used only by the idle thread, although  
    application threads can run at this priority if necessary."  
  doc ref/ecos-ref/ecos-kernel-overview.html#THE-SCHEDULER  
}  
  
}}CFG_DATA */  
  
#define CYGSEM_KERNEL_SCHED_MLQUEUE  
#undef CYGSEM_KERNEL_SCHED_BITMAP  
#define CYGNUM_KERNEL_SCHED_PRIORITIES 32
```

# プロパティ、ショートディスクリプション

Repository E:\Program Files\Cygnus Solutions\eCos - eCos Configuration Tool

File Edit View Build Tools Help

Configuration

- eCos HAL
- Infrastructure
- eCos kernel
  - Kernel interrupt handling
  - Exception handling
  - Kernel schedulers
    - Multi-level queue scheduler
    - Bitmap scheduler
    - Number of priority levels: 32
    - Scheduler timeslicing
    - Number of clock ticks between timeslices: 5
  - Counters and clocks
  - Thread-related options
    - Allow per-thread timers
    - Support optional name for each thread
    - Keep track of all threads using a linked list
    - Keep track of the base of each thread's stack
    - Support for per-thread data
    - Number of words of per-thread data: 6
    - Stack size for the idle thread: 2048
  - Synchronization primitives
  - Kernel instrumentation
  - Memory allocators
  - Source-level debugging support
  - Kernel APIs
- eCos uITRON Compatibility Layer
- C Library
- Math Library
- Common error code support
- Device drivers

**eCos 1.1 Documentation**

Getting Started with eCos

- Table of Contents
- Foreword
- Part I: Release Notes

Property	Value
Type	Integer
Value	32
Default Value	32
Macro	CYGNUM_KERNEL_SCHED_PRIORITIES
File	E:\Program Files\Cygnus Solutions\eCos\packages\kernel\v1_1\include\pkgconf\kernel.h
Defined at line	352
URL	E:\Program Files\Cygnus Solutions\eCos\doc\ref#ecos-ref#ecos-kernel-overview.html
Minimum	1
Maximum	32

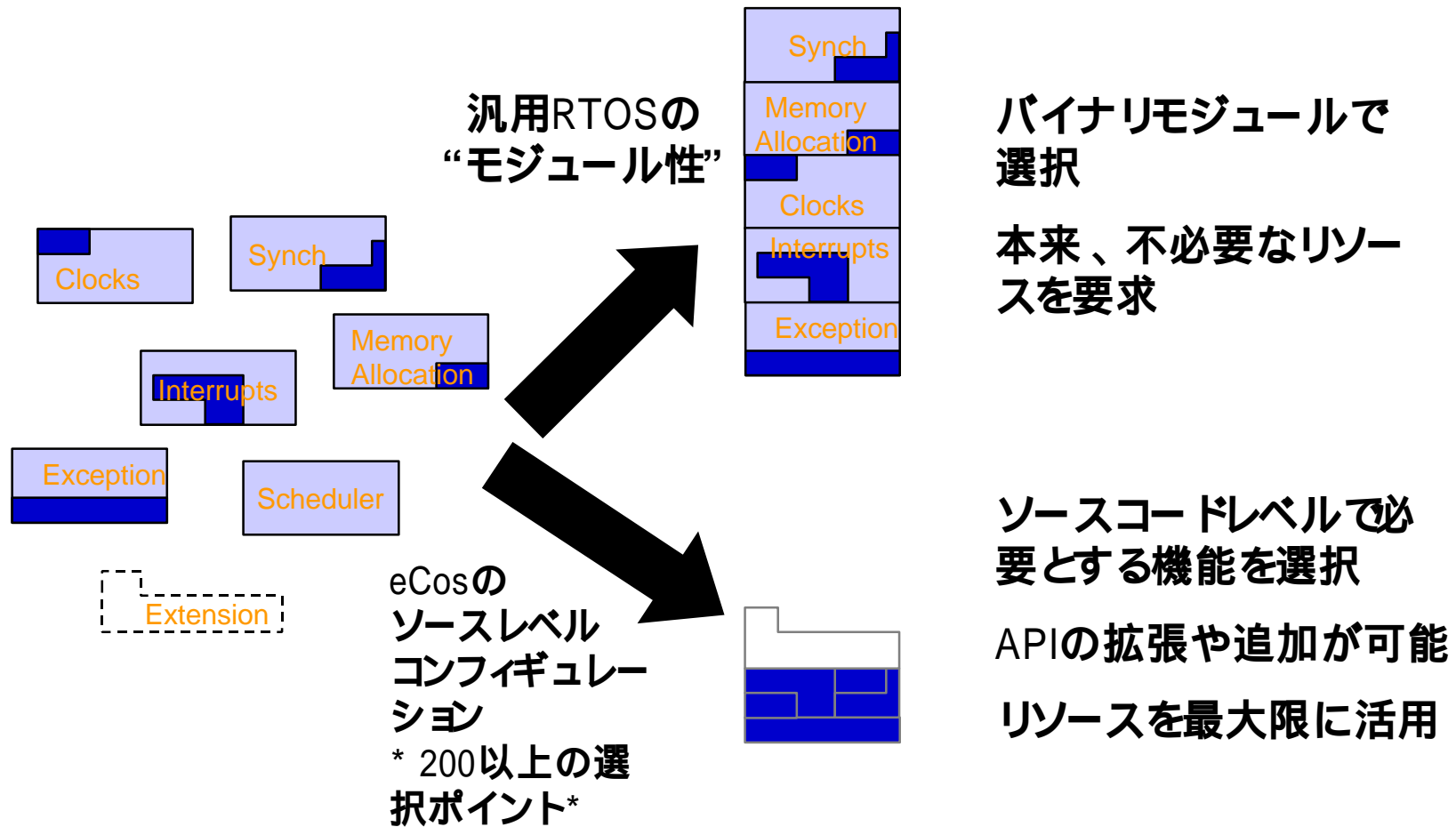
This option controls the number of priority levels that are available. For some types of scheduler including the bitmap scheduler this may impose an upper bound on the number of threads in the system. For other schedulers such as the mlqueue scheduler the number of threads is independent from the number of priority levels. Note that the lowest priority level is normally used only by the idle thread, although application threads can run at this priority if necessary.

プロパティ

ショートディスクリプション

Ready No failing rules E:\Program Files\Cygnus Solutions\eCos\doc\index.html

# アプリケーションに特化したコンフィギュレーション



# カーネルコンフィギュレーション例

- 松下製MN10300
- 10タスク
- 10セマフォ
- 15タイマ/アラーム
- 10メールボックス
- 20カウンタ
- デバッグサポート

イ  
ギ  
ユ  
シ  
ヨ  
ン

ROM (code)	9,437 Bytes
RAM (data)	948 Bytes
<hr/>	
TOTAL	10.4 Kbytes



# マイコンのサポート状況

東芝TX39、松下MN10300、モトローラPowerPC、富士通SPARClite、日立SH3、NEC VR4300、ARM7、  
半導体メーカーからの支持 東芝、富士通、ARM、モトローラ、松下、NEC、沖電気、PALMCHIP、STMicroelectronics

## eCosサポートパッケージ

- GNUPro 5ユーザー分のサポート
- eCos コンフィギュレーションツールの使用権とサポート
- eCos ランタイムソフトウェアのサポート
- 契約期間中のバージョンアップサービス

# V. 1.2.1 Host Toolの新機能

X86 Linux環境総合支援

Linuxの元でeCosのアプリケーションをネイティブ駆動

コンフィギュレーションツール

eCosのTest Case自動生成実行支援

コンフィギュレーションツール

メモリレイアウトエディタ支援

コンフィギュレーションツール

様々な操作性・機能性の改良

GDBのマルチスレッドサポート機能の改善

デバッグ中のスケ

ジューラロック機能支援

ダウンロードプロトコル改造

シリアル接続によるダウンロード時間を半減

# V.1.2.1 Runtime新機能

全サポートボードに対してeCosデバイスドライバ支援  
カーネルデバイスドライバAPIの標準化

ISO Cライブラリ支援、signals、date/time 関数支援

CygMon ROM monitor SPARClite評価ボード支援、イー  
サネットベースのダウンロードとデバッグ機能支援

新しいターゲットマイコンに対する移植、デバイスドライバ  
アーキテクチャ関連でドキュメント更新

テストプログラムの充実

多くの小規模な改造とバグ対策の実施

# ベンチマークテスト評価

- Board: Toshiba JMR3904 Evaluation Board
- CPU : Tmpr3904f 50MHz

出典:Sourceware.cygnus.com

- -----
- eCOS Kernel Timings
- Note: all times are in microseconds (.000001) unless otherwise stated
- Reading the hardware clock takes 0 "ticks" overhea... this value will be factored out of all other measurements
- Clock interrupt took 29.68 microseconds (45 raw clock ticks)

## Testing parameters:

- Clock samples: 32
- Threads: 24
- Thread switches: 128
- Mutexes: 32
- Mailboxes: 32
- Semaphores: 32
- Scheduler operations: 128
- Counters: 32
- Alarms: 32

**ボードやメモリによって性能は大きく左右されます。ここに掲載したものはほんの一例です。その他のマイコンによる実測結果は下記URLをご覧ください。**

<http://sourceware.cygnus.com/ecos/docs-1.2.1/guides/user-guides/sample-numbers.html>

# ベンチマークテスト評価

Ave	Min	Max	Var	Ave	Min	Function	単位:マイクロ秒
13.62	11.72	27.99	1.51	79%	54%	Create thread	
2.77	2.60	3.91	0.26	79%	79%	Yield thread [all suspended]	
3.31	2.60	6.51	0.27	83%	12%	Suspend [suspended] thread	
2.58	1.95	7.81	0.47	58%	37%	Resume thread	
4.94	4.56	11.07	0.60	95%	79%	Set priority	
0.71	0.65	1.95	0.10	95%	95%	Get priority	
14.97	14.32	25.39	0.87	95%	95%	Kill [suspended] thread	
2.25	1.95	9.11	0.57	95%	95%	Yield [no other] thread	
7.27	6.51	12.37	0.42	79%	16%	Resume [suspended low prio] thread	
2.28	1.95	7.16	0.51	95%	79%	Resume [runnable low prio] thread	
4.31	3.26	12.37	0.75	87%	79%	Suspend [runnable] thread	
2.17	1.95	7.16	0.42	95%	95%	Yield [only low prio] thread	
2.39	1.95	6.51	0.51	95%	58%	Suspend [runnable->not runnable]	
13.43	12.37	22.79	0.80	91%	91%	Kill [runnable] thread	
22.30	20.83	37.76	1.76	91%	91%	Resume [high priority] thread	
4.62	4.56	11.07	0.13	98%	98%	Thread switch	

# ベンチマークテスト評価

Ave	Min	Max	Var	Ave	Min	Function	単位:マイクロ秒
1.51	1.30	2.60	0.29	68%	68%	Scheduler lock	
2.36	1.95	3.26	0.31	61%	37%	Scheduler unlock [0 threads]	
2.39	1.95	5.21	0.32	62%	36%	Scheduler unlock [1 suspended]	
2.38	1.95	4.56	0.32	61%	37%	Scheduler unlock [many suspended]	
2.38	1.95	5.21	0.32	61%	37%	Scheduler unlock [many low prio]	
0.90	0.65	3.26	0.35	71%	71%	Init mutex	
2.48	1.95	8.46	0.50	50%	46%	Lock [unlocked] mutex	
2.83	2.60	9.11	0.42	93%	93%	Unlock [locked] mutex	
2.30	1.95	6.51	0.45	96%	65%	Trylock [unlocked] mutex	
1.99	1.30	5.86	0.24	84%	12%	Trylock [locked] mutex	
0.04	0.00	1.30	0.08	96%	96%	Destroy mutex	
42.40	42.32	44.92	0.16	96%	96%	Unlock/Lock mutex	

# ベンチマークテスト評価

Ave	Min	Max	Var	Ave	Min	Function	単位:マイクロ秒
1.44	1.30	5.86	0.28	96%	96%	Create mbox	
0.51	0.00	1.30	0.25	71%	25%	Peek [empty] mbox	
2.93	2.60	9.11	0.51	96%	78%	Put [first] mbox	
0.51	0.00	1.30	0.25	71%	25%	Peek [1 msg] mbox	
4.19	3.91	5.21	0.34	59%	59%	Put [second] mbox	
0.45	0.00	0.65	0.28	68%	31%	Peek [2 msgs] mbox	
3.28	2.60	10.42	0.45	65%	31%	Get [first] mbox	
3.34	2.60	9.77	0.40	78%	18%	Get [second] mbox	
2.69	1.95	9.11	0.40	78%	18%	Tryput [first] mbox	
2.75	1.95	7.81	0.32	93%	3%	Peek item [non-empty] mbox	
3.15	2.60	9.11	0.48	53%	43%	Tryget [non-empty] mbox	
2.22	1.95	6.51	0.41	96%	78%	Peek item [empty] mbox	
2.40	1.95	5.86	0.42	50%	46%	Tryget [empty] mbox	
0.47	0.00	0.65	0.26	71%	28%	Waiting to get mbox	
0.59	0.00	1.30	0.15	84%	12%	Waiting to put mbox	
4.01	3.26	10.42	0.40	81%	15%	Delete mbox	
26.18	26.04	30.60	0.28	96%	96%	Put/Get mbox	

# ベンチマークテスト評価

Ave	Min	Max	Var	Ave	Min	Function	単位:マイクロ秒
0.92	0.65	3.91	0.38	71%	71%	Init semaphore	
2.24	1.95	6.51	0.43	96%	75%	Post [0] semaphore	
2.32	1.95	7.16	0.48	96%	65%	Wait [1] semaphore	
2.03	1.30	5.86	0.24	90%	6%	Trywait [0] semaphore	
1.91	1.30	4.56	0.23	78%	18%	Trywait [1] semaphore	
0.77	0.00	1.95	0.30	65%	9%	Peek semaphore	
0.61	0.00	1.95	0.15	84%	12%	Destroy semaphore	
22.62	22.14	30.60	0.61	96%	62%	Post/Wait semaphore	
0.92	0.65	3.91	0.38	71%	71%	Create counter	
0.69	0.65	1.95	0.08	96%	96%	Get counter value	
0.41	0.00	1.30	0.33	56%	40%	Set counter value	
3.21	2.60	5.86	0.27	71%	21%	Tick counter	
0.65	0.00	3.26	0.16	84%	12%	Delete counter	

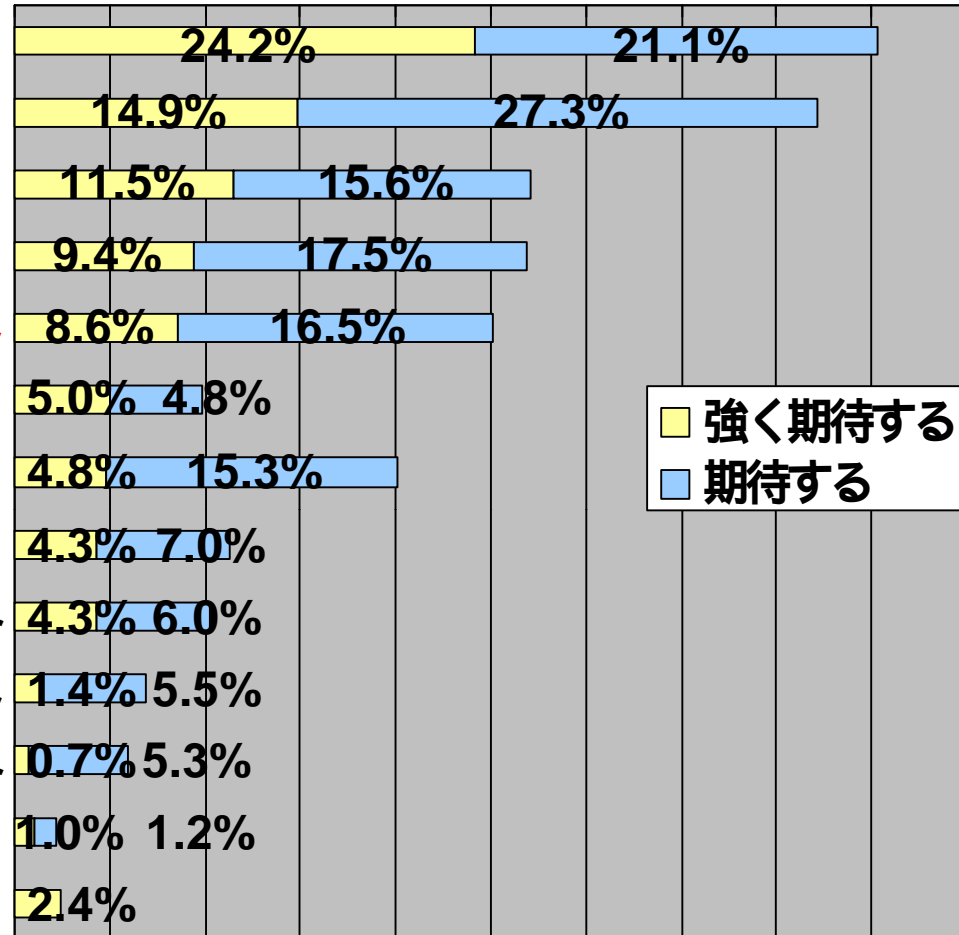


# ITRONの今後の取り組みに対する要望 (複数回答)

有効回答数 :417

eCosで実現

- 開発環境とのインターフェースの標準化
- ソフトウェア部品 (ミドルウェア) の  
インターフェイスの標準化
- C++/Java言語バインディングの標準化
- フリーのITRON仕様OS
- ネットワークサポート
- 応用分野を絞った標準化
- アプリケーション設計ガイドラインの作成
- 教育セミナーの開催
- ハードリアルタイムサポート
- マルチプロセッササポート
- フォールトトレランスサポート
- その他
- どれも期待しない



出典:1998.6 ITRON専門委員会報告より部分流用

0% 5% 10% 15% 20% 25% 30% 35% 40% 45% 50%

# eCosの位置づけと応用市場

## オープンソース、ロイヤリティフリーモデル

小規模メモリー、大量生産機器向け

Application Specific RTOSとして自由にコンフィギュレーション

カーナビゲーション、携帯電話、デジタルカメラ、デジタルTV、セットトップボックス、AV機器、プリンタ、FAX、複写機、OA機器、パソコン周辺機器、ゲーム等娯楽機器、FA工業機器、自動車、等々



# 今後のエンハンスの方向

年間少なくとも5～10ターゲット移植実行  
(GNUProでは年間25ターゲット以上実施)

ミドルウェアサプライヤのリクルート

デバッグ機能の充実(Profiler機能など)

IDEシステムへデバッグ環境の拡充

POSIXその他RTOS APIの拡張検討中

# eCosダウンロードサイト



<http://sourceware.cygnus.com/ecos>