

# システムソフトウェアの 現状と将来動向

慶應義塾大学環境情報学部  
慶應義塾大学大学院政策・メディア研究科  
徳田英幸

<http://www.ht.sfc.keio.ac.jp/>



# Outline

---

- システムソフトウェアに関する  
研究開発の現状
  - 事例 CMU Mach
  - 事例 Real-Time Linux
  - 問題点の整理
- 今後の研究開発と国の役割
- まとめ

# システムソフトウェアの位置づけ

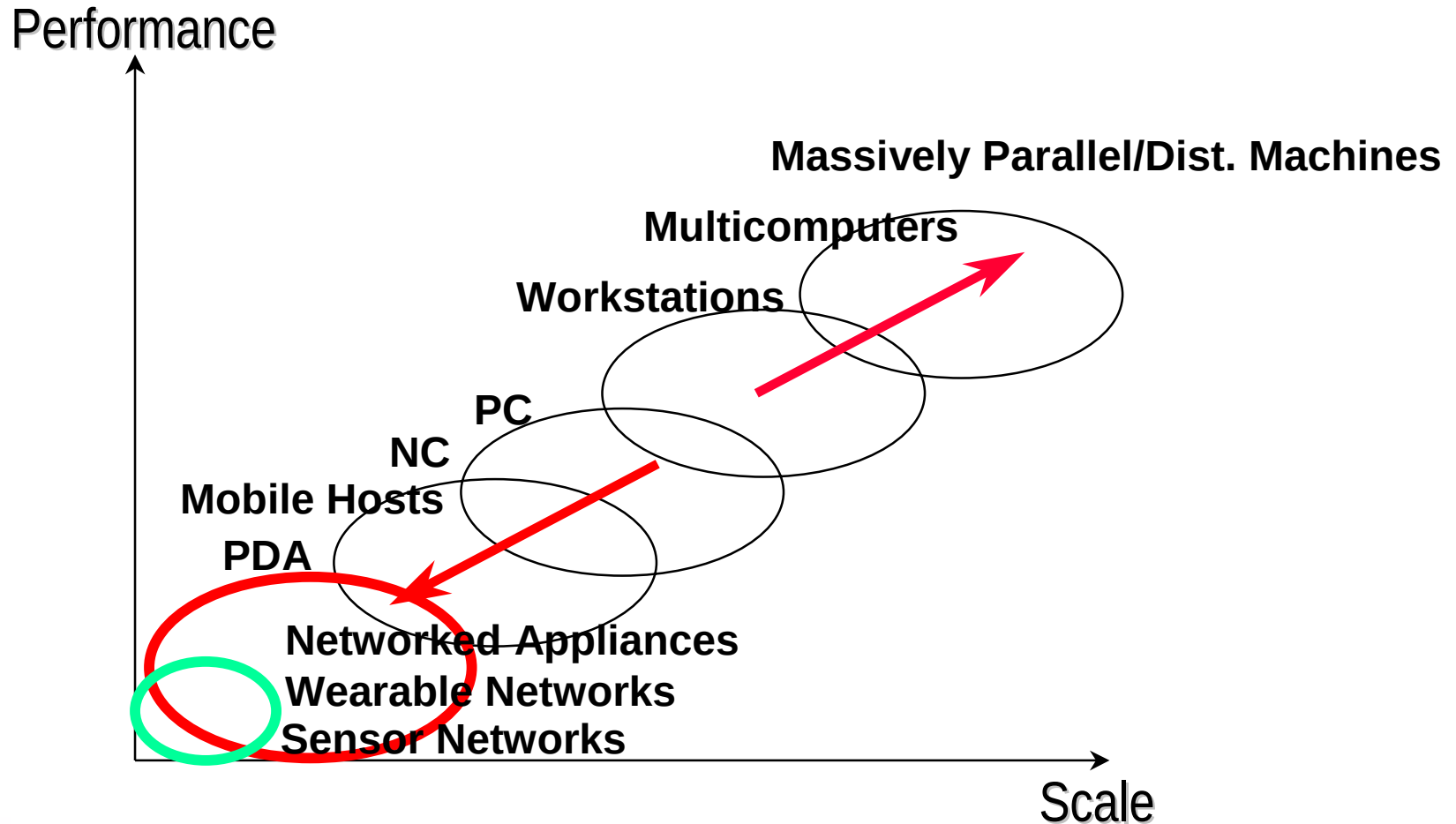


# A bit of history

---

- 事実1: 多くの優れたOS研究は、“オープン”であったBSD UNIXによるところが大きい。
- 事実2: 卓越したユーザたちが優れたOS技術を発展・進化させる。
- 事実3: 優れた研究者は、アクティブな研究開発コミュニティにて育成される。

# Which Direction?



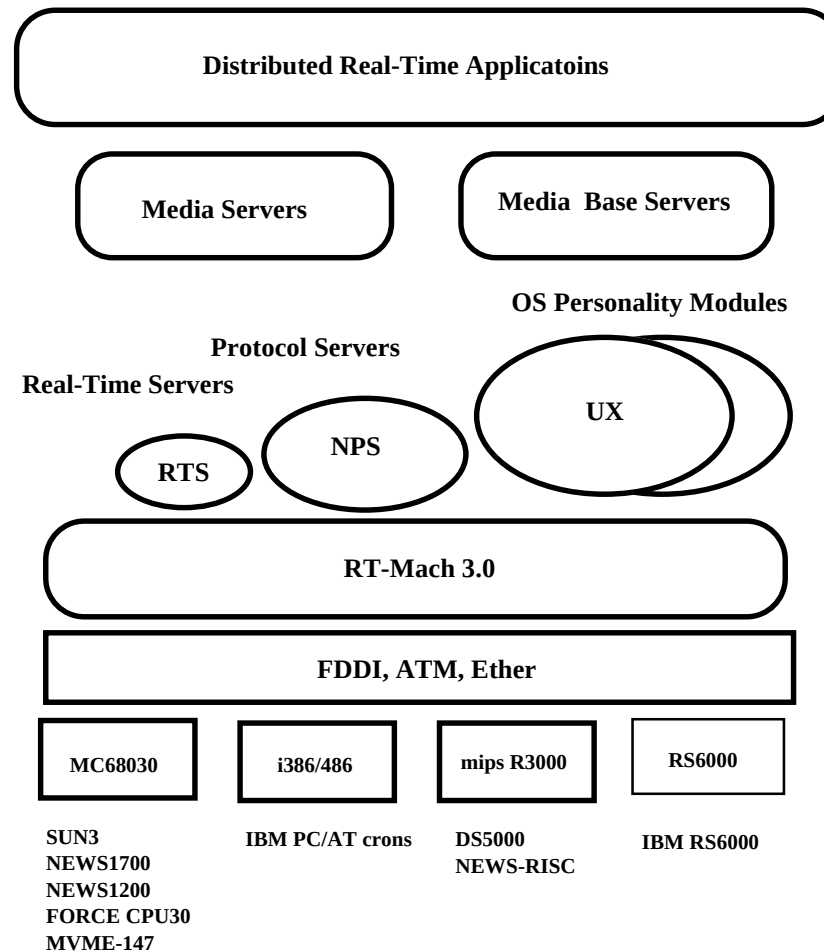
# 事例 : CMU Mach

---

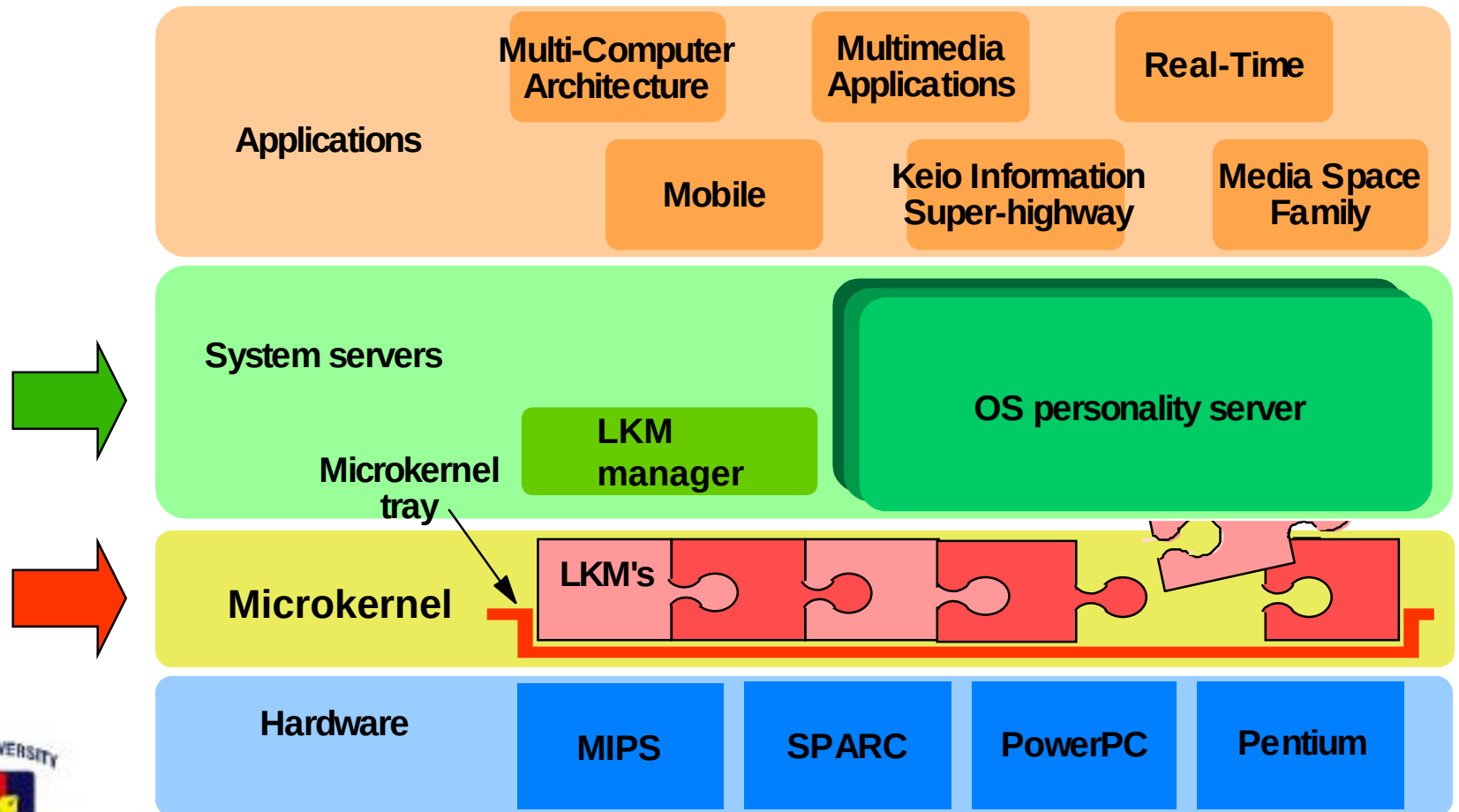
- CMU Mach
  - DARPA's Initiative
    - Distributed Common Kernel Workshop (1985)
    - Post 4.4BSD Unix + Parallel Processors
- Tech. Transfer
  - NeXT (Mach 2.5)
  - OSF-RI's Microkernel
  - SunOS, Chorus, others
  - MacOS X (Beta@2000)
    - (Mach Microkernel + Server Colocation+MacOS)



# マイクロカーネルアーキテクチャ (1)



# マイクロカーネルアーキテクチャ (2)





# 事例：リアルタイムLinux(出展 Linux Devices.com)

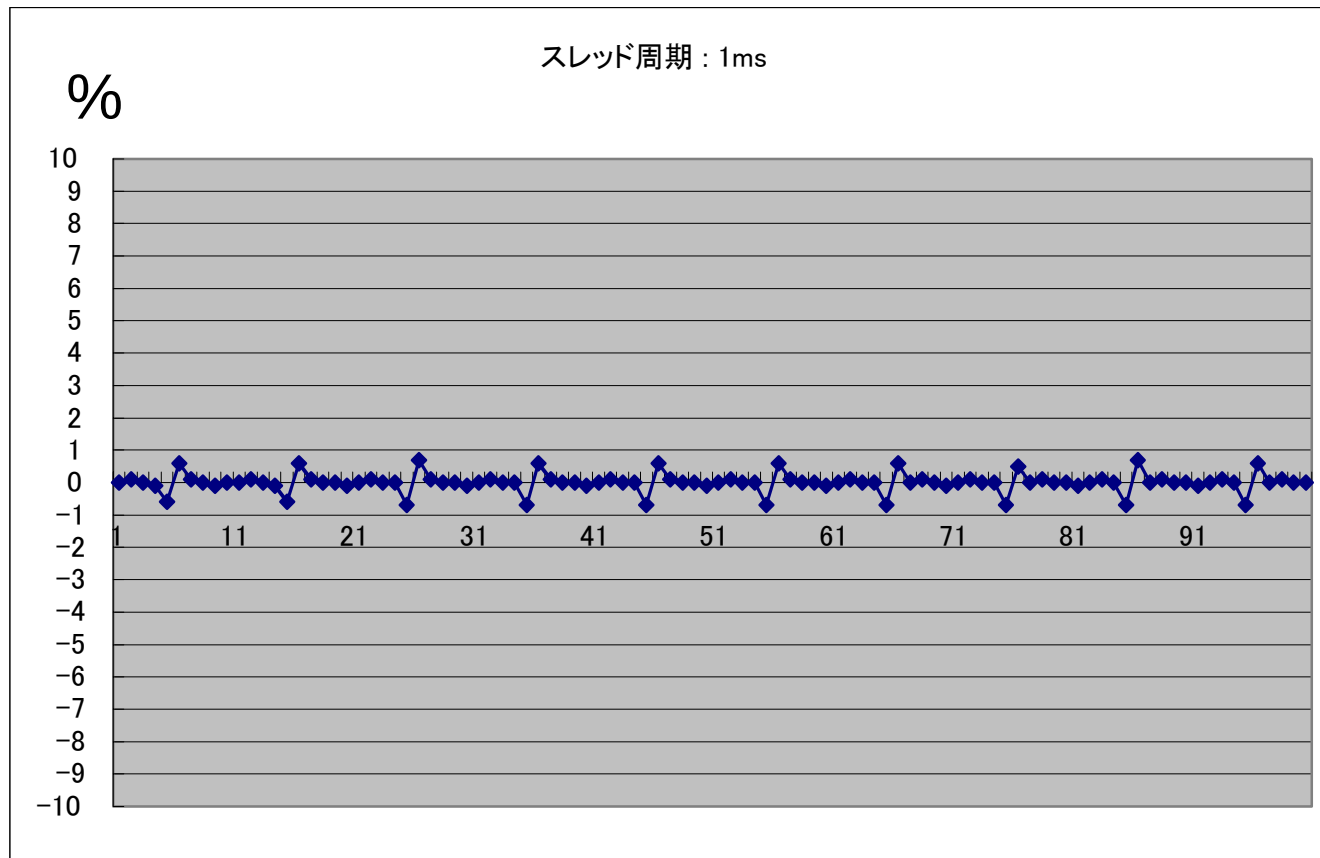
|                                                        | Commercial Real-time Distributions |                          |                              |                              | Open Source Real-time Projects |      |             |           |      |           |              |
|--------------------------------------------------------|------------------------------------|--------------------------|------------------------------|------------------------------|--------------------------------|------|-------------|-----------|------|-----------|--------------|
|                                                        | MontaVista<br>Hard Hat<br>Linux    | TimeSys<br>LinuxRT       | Lineo<br>Embedix<br>RealTime | REDSonix<br>RED-ICE<br>Linux | RTLinux                        | RTAI | RTAI + LXRT | RED-Linux | KURT | Linux-SRT | Molnar patch |
| Separate real-time kernel                              | with RTLinux                       | with RTAI                | with RTAI                    | with RTAI                    | ✓                              | ✓    | ✓           |           |      |           |              |
| User space programming                                 | ✓                                  | ✓                        | ✓                            | ✓                            | (2)                            |      | (1)         | ✓         | ✓    | ✓         | ✓            |
| Improved Linux kernel preemption                       | ✓                                  | within RK                |                              | ✓                            |                                |      |             | ✓         |      |           | ✓            |
| "Fully preemptible" Linux kernel                       | announced                          | announced                |                              |                              |                                |      |             |           |      |           |              |
| Precise scheduling, within Linux                       | ✓                                  | with RK,<br>without RTAI |                              | ✓                            |                                |      |             | ✓         | ✓    | ✓         |              |
| Includes solution for priority inversion, within Linux |                                    | with RK,<br>without RTAI |                              |                              |                                |      |             |           |      |           |              |
| "Fast" Interrupt response, < 1 mS                      | with RTLinux                       | with RTAI                | with RTAI                    | with RTAI                    | ✓                              | ✓    | ✓           |           |      |           |              |
| "Moderate" interrupt response, < 5 mS, within Linux    | ✓                                  | within RK                |                              | ✓                            |                                |      |             | ✓         |      |           | ✓            |
| Improved precision for timers, within Linux            |                                    | with RK,<br>without RTAI |                              | ✓                            |                                |      |             | ✓         | ✓    |           |              |
| NOTES: (1) - Limited (2) - Very limited                |                                    |                          |                              |                              |                                |      |             |           |      |           |              |

# 技術的問題点

---

- リアルタイム性
  - リアルタイム性能の保証
  - 消費電力問題
- 既存RTOS上のソフトウェア資産の有効利用
  - ハイブリッドアーキテクチャ
- 開発環境の高度化
  - デバック・検証ツール
- 動的再構成可能なライブラリとミドルウェア
  - Resource Reservation & QoS対応
  - モジュール間管理技術

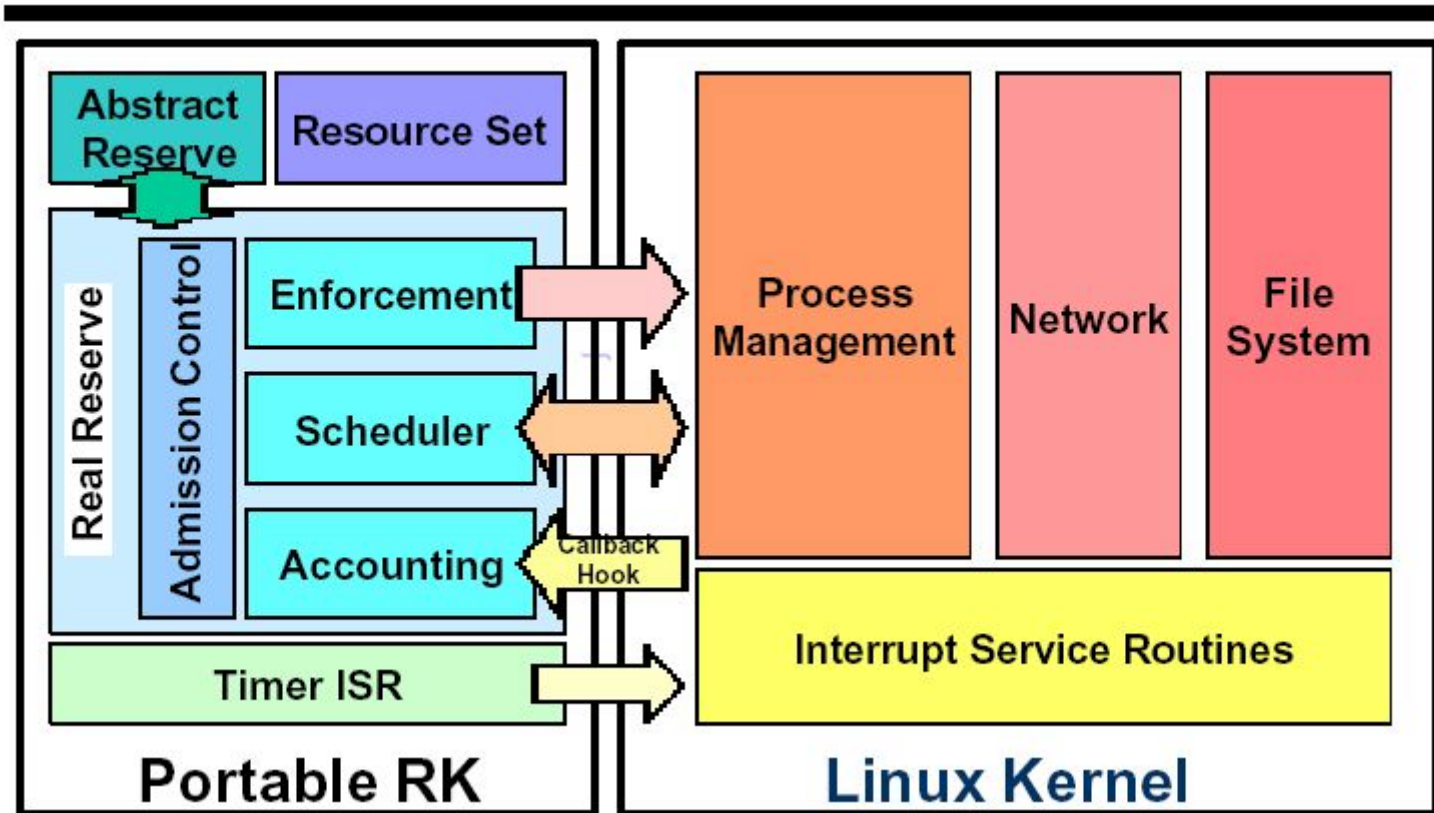
## 2.4カーネルベースでの実時間性能



# TimeSys Linux

出典: TimeSys社

User Space

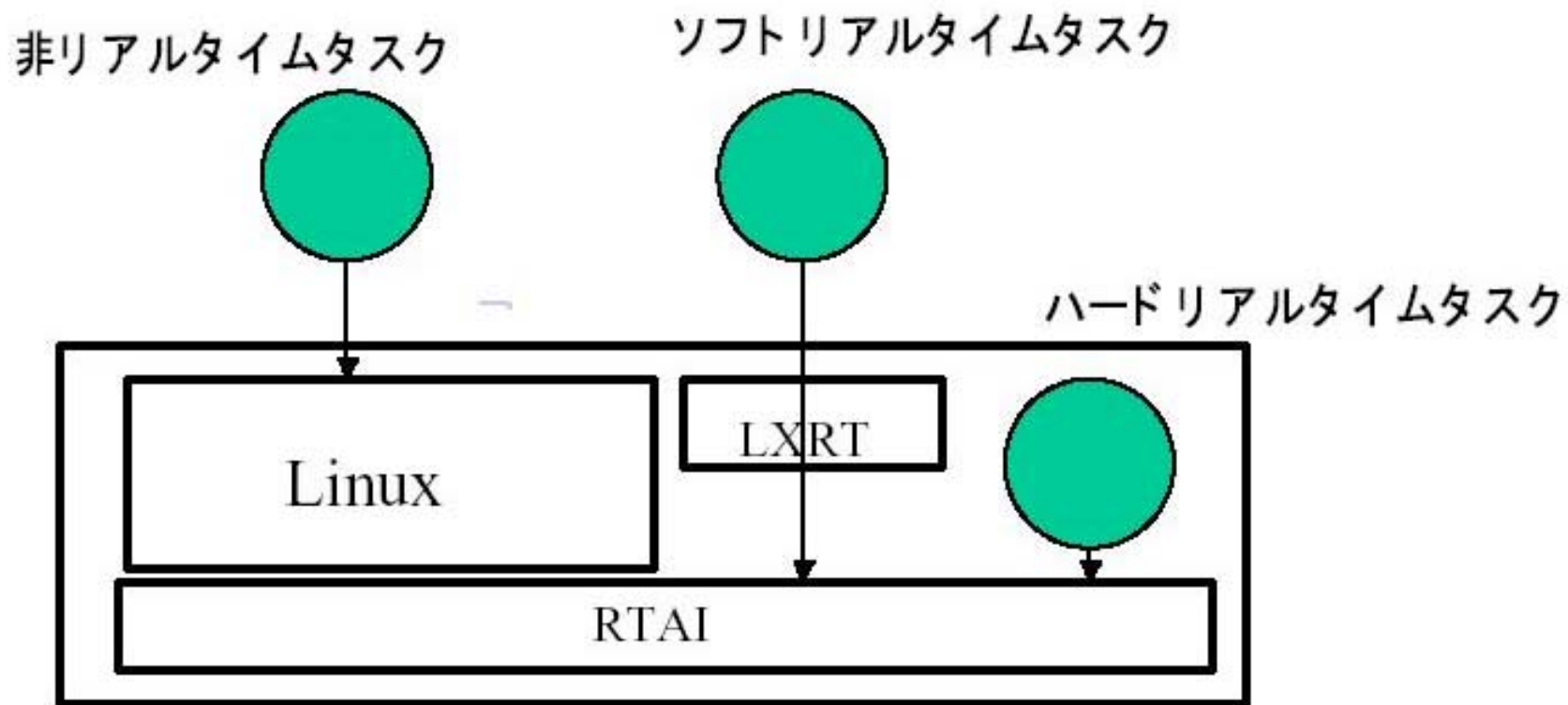


# Montavista HardHatLinux

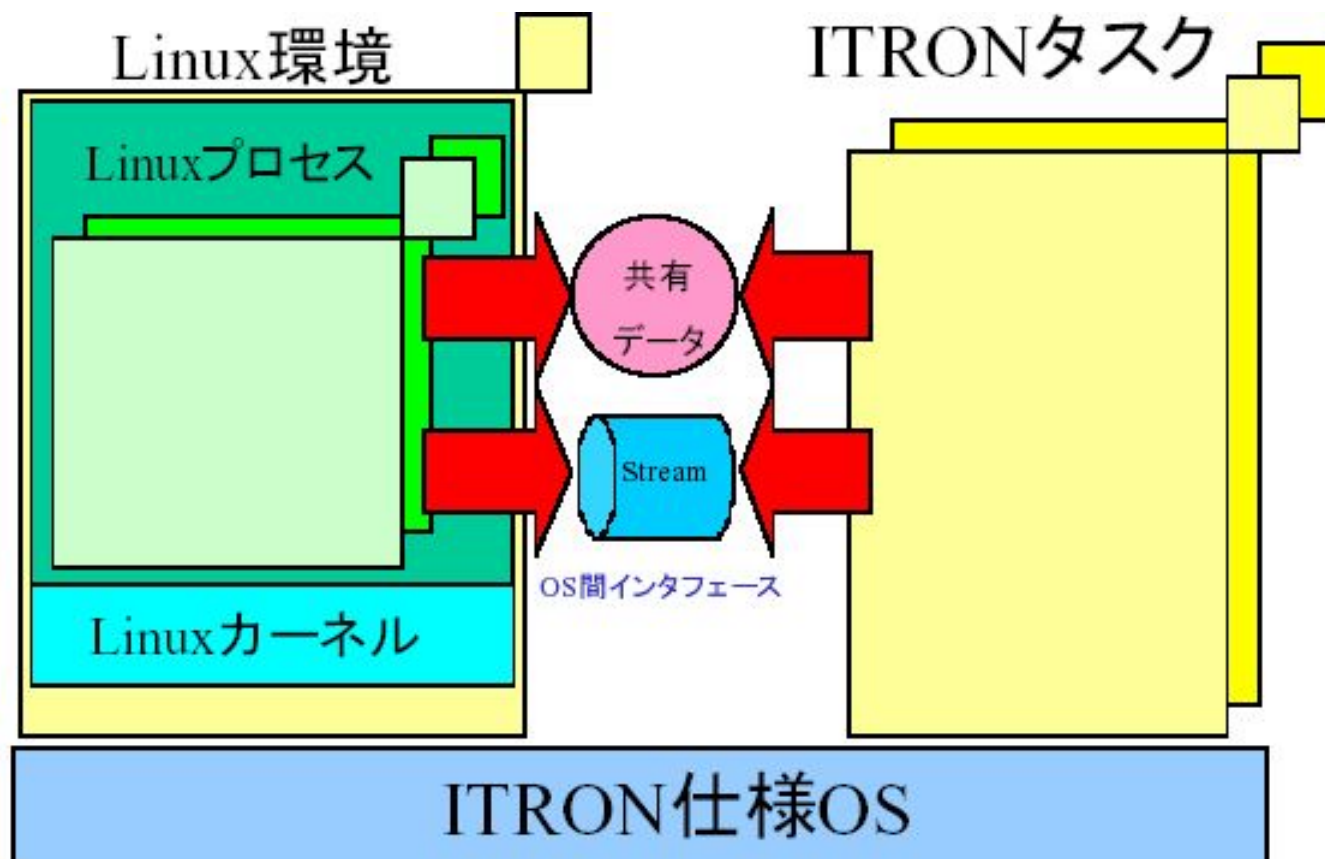
---

- Linux標準インタフェースを変更せず
- リアルタイム性の向上をめざす
  - リアルタイムスケジューラ
  - カーネルプリエンプタビリティ
  - カーネル内の非プリエンティブ部分の削除

# Lineo Embedix Linux



# Emblix: Japan Embedded Linux Consortium



# 問題点の整理

- OS研究開発コミュニティの育成
  - 「閉じた大学」から「開いた大学」へ
  - 実践的な問題の理解
- 基礎、応用、Tech Transferまで
  - 中長期的な戦略の建て直し
  - ユビキタスOS & ミドルウェア
- より科学的なソフトウェア作り
  - 「とにかく動けばいい」 から
  - 「ソフトウェアの科学的な作成」へ
  - 「HOW」が進化していない
  - 開発ツール、検証ツール、デバックツールの重要性



---

# 今後の研究開発と国の役割

# 国の役割 (1)

- OS研究開発コミュニティの確立
  - 中長期的な研究開発プラン
    - ユビキタスOS & Middleware開発
  - オープンソースの確立
  - 産官学の連携
  - 技術トランスファーの効率化
- オープンプラットフォームの研究開発
  - 共通に実験できるユビキタスプラットフォーム
    - ケイタイは閉じている！
    - オープンな実験プラットフォーム
- 開発・検証環境の高度化
  - 開発・検証効率の効率化

# 国の役割 (2)

- ユビキタスOSの研究開発
  - 動的適応可能ソフトウェアアーキテクチャ
    - 外的計算機環境と内的な計算機資源への適応
    - セキュリティへの対応
  - 状況依存型ソフトウェア
    - コンテキストウェア機構
    - パーソナライゼーション機構
  - ユビキタスミドルウェア支援
    - 動的再構成機構
    - プライバシー保護機構
  - ユビキタスアプリケーション支援
    - Service Roaming
    - Continuous Operations

---

# 付録：研究開発コミュニティの観点から

# 研究コミュニティ: Social Processの相違

---

- 大学
  - Research Univ. vs. 教育中心
  - 雇用システム
- 大学院システム
  - Stipend vs. No stipend
  - Research Stuff
- 大学の事務
  - Evaluation vs. No evaluation
  - Research support vs. A bit of support

# 研究コミュニティ: Social Processの相違 (2)

---

- RFPの文化/戦略
  - BAA vs. 公募方式
- プロポーザルの内容
  - Technical part + Budget part
  - Budget justification
  - 委託研究方式 vs. 請負方式
- 研究成果物の取扱い
  - PDS vs. 国庫
- Research meeting
  - PI meeting vs. 技術発表会

# 研究コミュニティ: Social Processの相違 (3)

---

- 企業との協調
  - Proposal vs. no proposal
  - Internal R&D
- Technology Transfer
  - PDS
  - ベンチャー vs. 産学連携
- Contract Monitor
  - Evaluation vs. No evaluation