

TOPPERSプロジェクトの現状と方向性 ～第3世代カーネル(ITRON系)の開発計画と状況～

2018年6月15日

高田 広章

NPO法人 TOPPERSプロジェクト 会長
名古屋大学 未来社会創造機構 教授
名古屋大学 大学院情報学研究科 教授
附属組込みシステム研究センター長
APTJ株式会社 代表取締役会長・CTO

Email: hiro@ertl.jp URL: <http://www.ertl.jp/~hiro/>

AGENDA

TOPPERSプロジェクトの概要と方向性(復習)

- ▶ これまでの開発の流れと現状
- ▶ 次の10年を見据えた活動指針(2011年度に策定)

TOPPERSプロジェクトの最新の成果

- ▶ この1年にリリースした開発成果

TOPPERS第3世代カーネル(ITRON系)の開発計画と状況

- ▶ 第3世代カーネル(ITRON系)の必要性, 仕様策定方針
- ▶ TOPPERS第3世代カーネル(ITRON系)統合仕様書
- ▶ TOPPERS/ASP3カーネル, TOPPERS/HRP3カーネル
- ▶ TECSの活用, Ruby版コンフィギュレータの適用
- ▶ 今後の開発計画

TOPPERSプロジェクトの方向性

- ▶ 組込みソフトウェア開発効率化のためのTOPPERS

TOPPERSプロジェクトの 概要と方向性（復習）

TOPPERSプロジェクトとは?

TOPPERS = Toyohashi Open Platform for
Embedded and Real-Time Systems



プロジェクトの活動内容

- ▶ ITRON仕様の技術開発成果を出発点として、組み込みシステム構築の基盤となる各種の高品質なオープンソースソフトウェアを開発するとともに、その利用技術を提供

組み込みシステム分野において、Linuxのように広く使われるオープンソースOSの構築を目指す!

プロジェクトの推進主体

- ▶ 産学官の団体と個人が参加する産学官民連携プロジェクト
- ▶ 2003年9月にNPO法人として組織化
- ▶ それ以前は、名古屋大学(2002年度までは豊橋技術科学大学)高田研究室を中心とする任意団体として活動

TOPPERSプロジェクトの狙い

決定版のITRON仕様OSの開発 **完了**

- ▶ ITRON仕様がかかえる過剰な重複投資と過剰な多様性の問題を解決(または軽減)

次世代のリアルタイムOS技術の開発

- ▶ 組み込みシステムの要求に合致し, ITRONの良さを継承する次世代のリアルタイムOS技術を開発

Linuxと類似のOSをもう1つ作っても意味がない!

- ▶ オープンソースソフトウェア化により産学官の力を結集

組み込みシステム開発技術と開発支援ツールの開発

- ▶ 高品質な組み込みシステムの効率的な開発を支援

組み込みシステム技術者の育成への貢献

- ▶ オープンソースソフトウェアを用いた教育コースや教材を開発し, それを用いた教育の場を提供

開発成果物の主な利用事例



エスクード (スズキ)



スカイラインハイブリッド (日産)



IPSiO GX e3300 (リコー)



H-IIIB (JAXA)



Cell³iMager duos
(SCREEN
ホールディングス)



OSP-P300
(オークマ)



SoftBank
945SH
(シャープ)



UA-101 (Roland)



PM-A970(エプソン)

次の10年を見据えた活動指針 (2011年度に策定)

Smart Futureのための組込みシステム技術

- ▶ 組込みシステム技術を、持続可能なスマート社会の実現 (Smart Future) のための重要な要素技術の1つと位置づけ、その研究開発と普及に取り組む
- ▶ それに向けての研究開発課題
 - ▶ Safety & Security
 - ▶ Ecology (高エネルギー効率)
 - ▶ Connectivity

コンソーシアムによるオープンソースソフトウェア開発

- ▶ 同じ技術に関心を持つプロジェクトメンバによりコンソーシアムを結成し、複数組織の協力によりソフトウェアを開発
- ▶ 開発したソフトウェアは、TOPPERSプロジェクトからオープンソースソフトウェアとして公開

重点的に取り組んでいるテーマ

※ SPF:ソフトウェア
プラットフォーム

次世代のリアルタイムカーネル技術

- ▶ TOPPERS第3世代カーネル(ITRON系)
- ▶ 車載システム向けRTOS(AUTOSAR OS仕様からの発展)

ソフトウェア部品化技術

- ▶ TECS(TOPPERS組込みコンポーネントシステム)

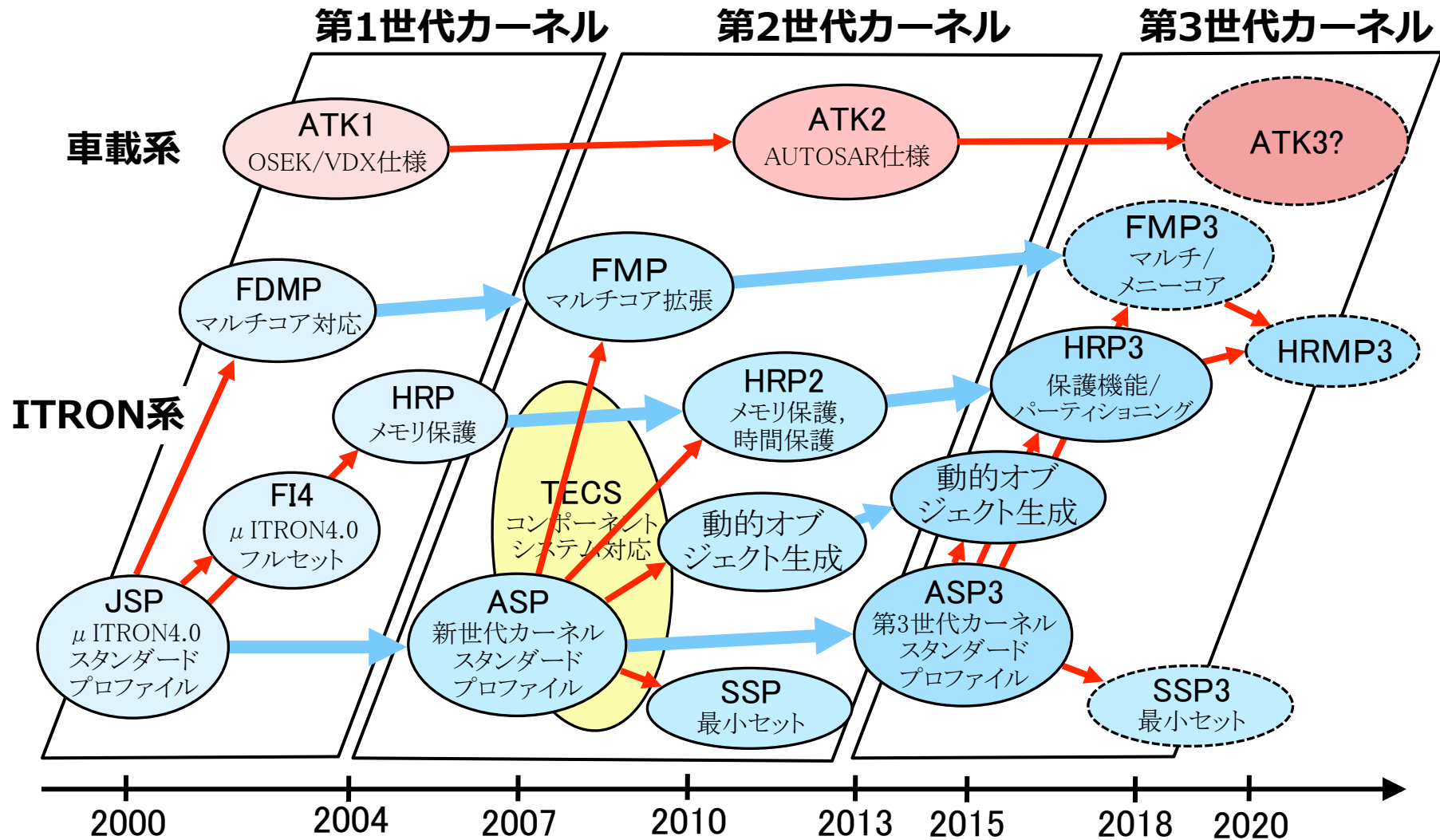
組込みシステム向けSPFと開発支援ツール

- ▶ 車載制御システム向けSPF(AUTOSAR仕様ベース)
- ▶ 仮想化技術(SafeG, A-SafeG), ホームネットワーク
- ▶ 宇宙機向けSPF(SpaceWire OS), BASE PLATFORM
- ▶ 開発支援ツール(シミュレータ, 可視化ツール)

技術者育成のための教材開発

- ▶ プラットフォーム技術者の育成
- ▶ ETロボコン向けSPFと教材の提供

TOPPERSカーネル開発ロードマップ



TOPPERSプロジェクトの最新の成果

この1年にリリースした主な開発成果

! 特定ターゲット向けの簡易パッケージは除く
2017年6月～12月 一般公開

6月7日 TINET 1.7

6月16日 TECSジェネレータ V1.5.0

6月23日 MDCOM (MultiDomain Communication Module)
Release 1.0.0 (最初の一般公開)

- ▶ ドメイン間 (RTOSと汎用OSの間) の効率的な同期・通信

- ▶ 共有メモリチャネルとFIFOチャネル

7月21日 TOPPERS/ASP3カーネル Release 3.2.0

7月22日 TOPPERS第3世代カーネル統合仕様書 Release
3.1.0

12月23日 TOPPERS BASE PLATFORM対応版基礎1,2セミナー教材 (最初の一般公開)

2018年1月～2018年5月 一般公開

- 1月18日 mruby on GR-PEACH+TECS (最初の一般公開)
- 4月25日 TOPPERS/ASP3カーネル Release 3.3.0
- 同日 TOPPERS/HRP3カーネル Release 3.0.0 (最初の一般公開)
- 同日 TOPPERS第3世代カーネル統合仕様書 Release 3.2.0
- 4月25日 TOPPERS/EV3RT Version Beta 7.1
- 4月29日 TOPPERS第3世代カーネル統合仕様書 Release 3.2.1
- 同日 TOPPERS第3世代カーネル統合仕様書 Release 3.2.1 リッチテキスト版(テスト版)
- 5月9日 TECSジェネレータ V1.6.1
 - ▶ TECSInfoの正式サポート
 - ▶ TOPPERS/HRP3カーネル対応

2018年6月 一般公開

6月3日 mruby on EV3RT+TECS beta 2.1.0

6月4日 TOPPERS BASE PLATFORM(ST) および
TOPPERS BASE PLATFORM(CV) (最初の一
般公開)

- ▶ 各種のデバイスドライバ, USBミドルウェア (ホスト／デバイス) などの各種のミドルウェアを含む

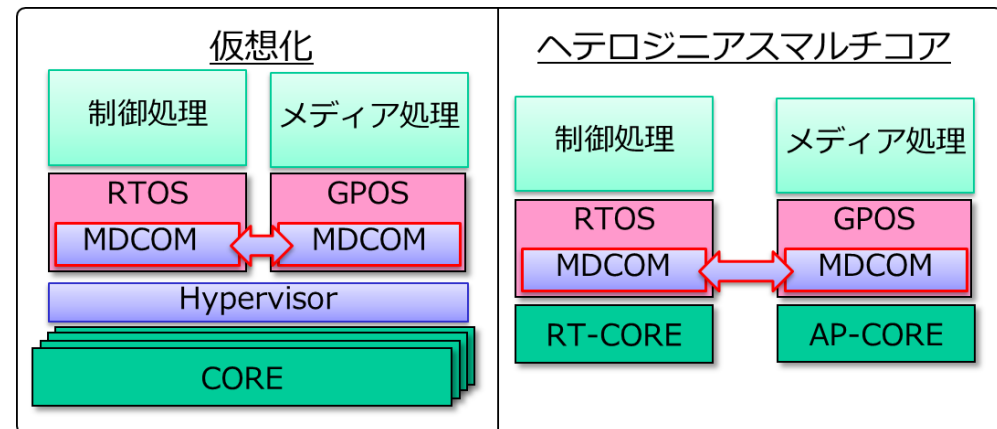
2017年度内のContributed Softwareへの新規アップロード

- ▶ 動的生成版ASP + TINETのRuby版CFG
 - ▶ Ruby版コンフィギュレータで, 動的生成機能拡張したASPカーネルとTINET 1.7をコンフィギュレーションできるようにしたもの
- ▶ TOPPERS/ASP3版ECNL

MDCOM (MultiDomain Communication Module)

ドメイン間通信をサポート

- ▶ RTOS (TOPPERSカーネル) 上のタスクと汎用OS (Linux) 上のプロセス間での通信をサポート
- ▶ 仮想化とヘテロジニアスマルチコアの両方で使用可能



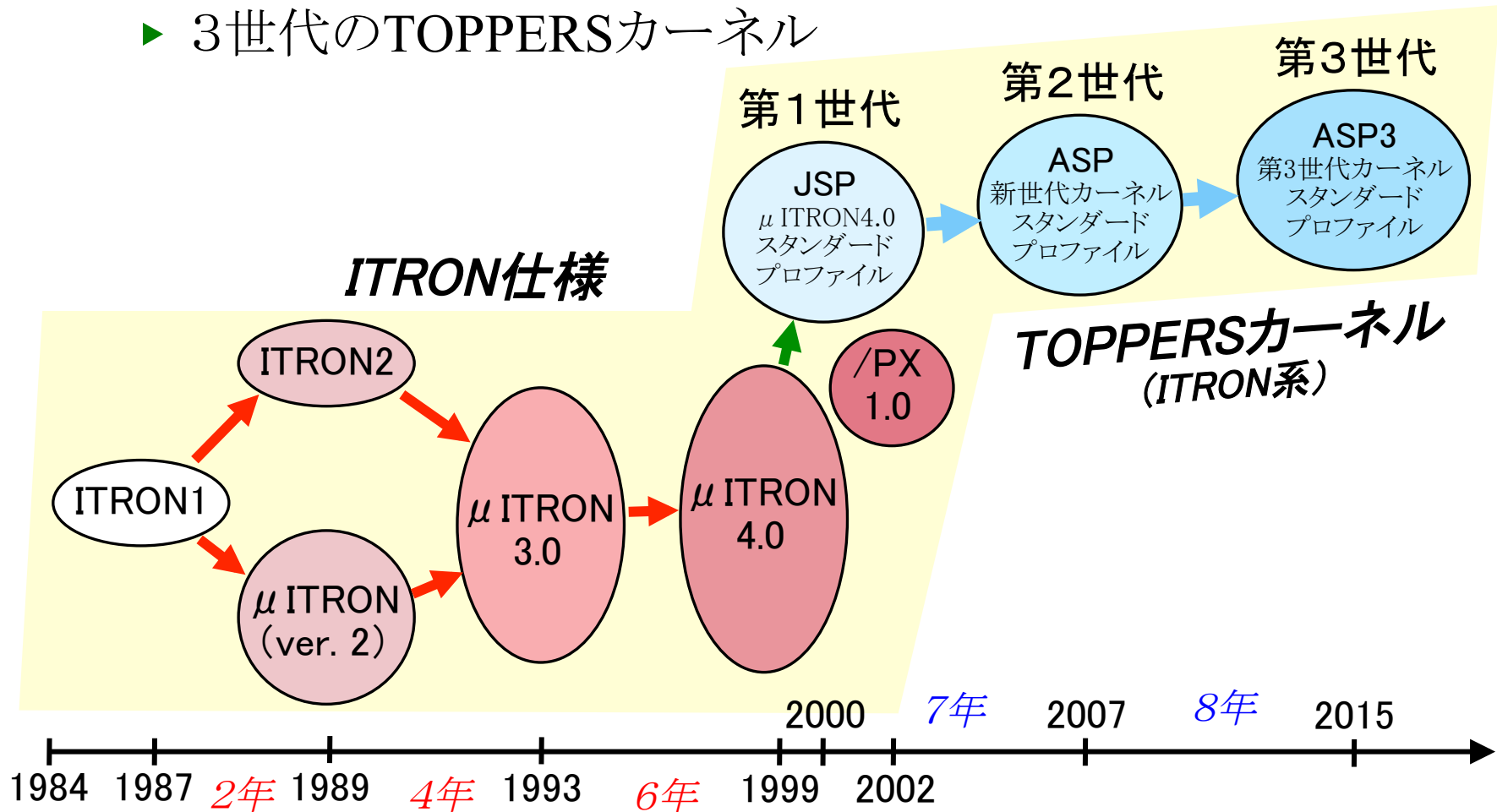
2種類の通信機構

- ▶ SMEMチャンネル: 共有メモリを提供するチャンネル. 排他制御のためのロックも提供
- ▶ FIFOチャンネル: FIFO通信のためのチャンネル. イベント通知機能も提供

TOPPERS第3世代カーネル (ITRON系)の開発計画と状況

ITRON仕様～TOPPERSカーネル：30年超の歴史

- ▶ 4世代のITRON仕様
- ▶ 3世代のTOPPERSカーネル



リアルタイムカーネル開発の流れ

! 高信頼性・安全性・リアルタイム性を追求

第1世代のリアルタイムカーネル

- ▶ μ ITRON4.0仕様準拠 + α のリアルタイムカーネル
 - ▶ TOPPERS/JSP, FI4, FDMP, HRP
- ▶ OSEK/VDX OS仕様準拠のリアルタイムカーネル
 - ▶ TOPPERS/ATK1

第2世代のリアルタイムカーネル

- ▶ TOPPERS新世代カーネル (ITRON仕様からの発展)
 - ▶ TOPPERS/ASP, FMP, HRP2, SSP
- ▶ AUTOSAR OS仕様ベースのリアルタイムカーネル
 - ▶ TOPPERS/ATK2 (SC1, SC3, SC1-MC, SC3-MC, ...)

第3世代のリアルタイムカーネル

- ▶ TOPPERS第3世代カーネル (ITRON系)
 - ▶ TOPPERS/ASP3, TOPPERS/HRP3, ...

TOPPERS第3世代カーネル (ITRON系) の必要性

求められている/求められつつある技術・機能

- ▶ 機能安全からの要求に応えられるパーティショニング **Safety**
- ▶ ティックレスの高分解能時間管理と外部時刻同期 **Connectivity**
- ▶ マルチコアにおける動的ロードバランシング **Ecology**
- ▶ メニーコアプロセッサへの対応 (… 今後の課題) **High performance**

一方, 廃止すべきと考えられる機能もある

- ▶ タスク例外処理機能 … 複雑さと比べて使われていない
- ▶ メールボックス … 安全性の高い機能で代替

TOPPERS第3世代カーネル (ITRON系) へ

- ▶ 新しい世代のリアルタイムカーネルと位置付けた方が, 大胆な仕様変更 (特に, 機能の削除) が可能

TOPPERSカーネル開発への想い

読みやすく, 改造しやすいカーネルとしたい

- ▶ オープンソースソフトウェアの品質を向上させる上で重要
 - ▶ 多くの方に読んで欲しい
- ▶ ユーザがニーズに応じて改造できるようにしたい
 - ▶ 様々なユーザニーズに, 1つのソフトウェア(ソースコード)で応えるのは難しい

お手本としてのTOPPERSを目指したい

- ▶ ソフトウェア技術者の人材育成のためには, 良いお手本が必要
 - ▶ プログラムを読ませる前に書かせていないか?
 - ▶ 国語教育は, 良い文章を読むことから始まる
- ▶ TOPPERSは, 質の高いソフトウェアとは何か, 良いドキュメントとは何かを学ぶための良いお手本を目指したい

第3世代カーネル (ITRON系) の仕様策定方針

基本的な設計方針 … 第2世代カーネルを踏襲

- ▶ μ ITRON4.0仕様をベースに拡張・改良を加える
- ▶ ソフトウェアの再利用性を重視する
- ▶ 高信頼・安全なシステム構築を支援する
- ▶ アプリケーションシステム構築に必要な機能は積極的に取り込む

特に留意した設計方針(これまでも暗黙に存在)

- ▶ システム／アプリケーションによって要求が異なる機能は、ミドルウェア等で実現することとし、カーネルにはその実現に必要な最低限の機能を導入する
 - ▶ ポリシーとメカニズムの分離の考え方
- ▶ 実装方法を合わせて検討し、オーバヘッドの大きい仕様を避ける

第2世代カーネルからの仕様変更

(1) 時間パーティショニング機能の導入

- ▶ 機能安全におけるソフトウェアパーティショニングの実現を容易にするための機能

(2) ティックレスの高分解能時間管理と外部時刻同期

- ▶ カーネルが管理する時間の分解能をミリ秒からマイクロ秒に変更
 - ▶ タイムティックを使わない実装に変更
- ▶ 外部時刻との同期のための機能を導入

(3) マルチコアにおける動的負荷分散機能への対応

- ▶ マルチコアプロセッサにおけるロードバランシングを実現するために以下の機能を導入
 - ▶ タスクとスケジューリング状態の参照機能
 - ▶ サブ優先度機能

(4)仕様のスリム化・シンプル化(主なもの)

- ▶ タスク例外処理機能の廃止とタスク終了要求機能の導入
 - ▶ 必要性が低いにもかかわらず, ターゲット依存部(特に保護機能対応の場合)の実装負担が大きいタスク例外処理機能と待ち禁止状態を廃止
- ▶ 非タスクコンテキスト専用のサービスコールの廃止
- ▶ メールボックス機能の廃止
 - ▶ 保護機能対応では不適切な機能

(5)ミューテックスの標準機能化

(6)その他の改良(主なもの)

- ▶ タイムイベント処理機能の見直し
- ▶ 起動要求をキューイングしないタスクの追加
- ▶ 保護ドメインに対するアクセス許可ベクタの新設
- ▶ 保護ドメイン毎の標準メモリアリケーション

TOPPERS第3世代カーネル(ITRON系)統合仕様書

ドキュメントの位置付け

- ▶ TOPPERS第3世代カーネル(ITRON系)に属する一連のリアルタイムカーネルの仕様を, 統合的に記述したもの
 - ▶ 各カーネルの仕様の相違がわかりやすい. 上位のカーネルへの乗り換えを考慮

ドキュメントのサイズ

- ▶ プレインテキスト形式: 約19,800行 (図を含まない)
- ▶ PDF形式 (公開版): 約400ページ (図を含む)

タグの付与

- ▶ カーネルの実装に対する要求事項等に, トレーサビリティのためのタグを付与
 - ▶ タグの形式: NGKIxxxx (xは数字), ASPSxxxxなど

プレーンテキストによる記述の利点・欠点

- ▶ 利点
 - ▶ 差分を取ることが容易
 - ▶ ファイルを分割するのも自由
 - ▶ ツールが軽い(大きいファイルでも容易に扱える)
- ▶ 問題点
 - ▶ 体裁(かっこ)が悪い
 - ▶ リンクがはれない(PDFの目次の機能が使えない, など)
 - ▶ 図が離れていて不便

リッチテキスト化(トライアル中)

- ▶ “ルール化されたプレーンテキスト” の考え方の導入
- ▶ リッチテキスト(現在は, AsciiDocを利用)に変換するツール(Rubyスクリプトで1000行以下)を開発
- ▶ テストリリース中. 評判が良ければ, こちらに移行

第3世代カーネル (ITRON系) の実装

! カーネルのバージョン番号を「3」に統一
TOPPERS/ASP3カーネル

- ▶ TOPPERS第3世代カーネルの出発点

TOPPERS/HRP3カーネル

- ▶ ASP3カーネルに保護/パーティショニング機能を追加し、さらに高い信頼性・安全性を要求されるシステムに対応

TOPPERS/FMP3カーネル(開発中)

- ▶ ASP3カーネルをマルチ/メニーコアプロセッサ向けに拡張

TOPPERS/HRMP3カーネル(計画中)

- ▶ HRP3カーネルとFMP3カーネルを統合

TOPPERS/SSP3カーネル(開発中)

- ▶ ASP3カーネルをベースに機能を絞り込む

TOPPERS/ASP3カーネルの概要

ASP3カーネルの位置付け

- ▶ TOPPERS第3世代カーネルの出発点
- ▶ TOPPERS/ASPカーネルを拡張・改良したもの

ASPカーネルに追加した機能(基本パッケージでサポート)

- ▶ タスク終了要求機能
- ▶ ティックレスの高分解能時間管理, システム時刻の調整機能, システム時刻の参照／設定機能
- ▶ ミューテックス機能 … 基本機能に格上げ
- ▶ 周期通知／アラーム通知機能 … 周期ハンドラ／アラームハンドラ機能を拡張

ASPカーネルから削除した機能

- ▶ タスク例外処理機能
- ▶ メールボックス機能

周辺ツールの活用・変更

- ▶ TECSの活用
 - ▶ システムサービス(システムログ機能など)とデバイスドライバの構築にTECSを活用
 - ▶ カーネルを利用するだけであれば, TECSを勉強する必要はない
- ▶ Ruby版コンフィギュレータの適用

拡張パッケージでサポート

- ▶ ドリフト調整機能
- ▶ メッセージバッファ機能
- ▶ オーバランハンドラ機能
- ▶ タスク優先度拡張
- ▶ 制約タスク
- ▶ サブ優先度機能
- ▶ 動的生成機能

ASP3カーネルの実装設計方針

「TOPPERS/ASP3カーネル 設計メモ」より

- (1) ソースの読みやすさ, 改造しやすさを重視する
 - ▶ オープンソースソフトウェアの品質を向上させるために重要な特性
 - ▶ システムの要求にあわせたチューニングが行いやすい
- (2) 新しいターゲットシステムへのポーティングが容易な構造とする
- (3) 検証が容易な構造とする
- (4) 実行性能とメモリ使用量に配慮する
- (5) スケーラビリティに配慮する
 - ▶ 様々な規模のシステムに使えるようにする
 - ▶ 使用しない機能によりメモリ使用量が増えないように
- (6) タイムティックを用いない時間管理(ティックレスカーネル)

ソフトウェアのサイズとトレーサビリティ

カーネル本体

- ▶ ソースコードのサイズ(コメント等を含む)
 - ▶ ターゲット非依存部のCソースコード:約11,000行(コメントを除くと半分程度?)
 - ▶ ターゲット依存部のC＋アセンブリ言語ソースコード:約8,000行(ターゲットにもよる, コメント比率が高い)
- ▶ バイナリのサイズ:約32KB (ARMプロセッサ)
- ▶ ソースコードと統合仕様書の記述が直接対応する場合には, 統合仕様書のタグを参照

テストプログラム

- ▶ 45件のテストプログラム, 4件のコンフィギュレータテスト
- ▶ 統合仕様書の記述に対応するテストでは, 統合仕様書のタグを参照

ドキュメントのサイズとトレーサビリティ

ユーザ向けドキュメント

- ▶ ユーザ向けドキュメントのサイズ
 - ▶ ユーザーズマニュアル: 約2,700行
 - ▶ 仕様概要, 機能拡張・チューニングガイド, マイグレーションガイド, バージョン履歴: 合計約2,500行
 - ▶ 他に, ターゲット依存部のドキュメントがある

開発者向けドキュメント

- ▶ 開発者向けドキュメントのサイズ
 - ▶ ポーティングガイド: 約4,500行
 - ▶ 設計メモ: 約1,100行 ... あくまでもメモ, 不完全
 - ▶ コンフィギュレータ仕様: 約1,500行
 - ▶ 他に, ターゲット依存部のドキュメントがある
- ▶ 設計メモとコンフィギュレータ仕様では, 必要に応じて, 統合仕様書のタグを参照

TOPPERS/HRP3カーネルの概要

HRP3カーネルの位置付け

- ▶ ASP3カーネルに保護機能／パーティショニング機能を追加したもの（ASP3カーネルの上位互換）
- ▶ TOPPERS/HRPカーネル, HRP2カーネルの改良・拡張版

ASP3カーネルに対する追加機能

- ▶ メモリ保護機能
 - ▶ 自動メモリ配置（HRP2カーネルと同様）と手動メモリ配置（HRPカーネルと同様）の両方をサポート
- ▶ オブジェクトアクセス保護機能
- ▶ 時間パーティショニング機能
- ▶ 拡張サービスコール機能
- ▶ メッセージバッファ機能（ASP3では拡張パッケージ）

HRP2カーネルに対する追加／削除機能

- ▶ ASPカーネルからASP3カーネルで追加／削除した機能
- ▶ 〔追加〕時間パーティショニング機能
- ▶ 〔追加〕手動メモリ配置
- ▶ 〔削除〕待ち禁止状態とそれ进行操作する機能
- ▶ 〔削除〕オーバランハンドラ機能(拡張パッケージに)

周辺ツールの活用・変更

- ▶ TECSの活用
 - ▶ *ASP3カーネルと同一のシステムサービス(システムログ機能など)とデバイスドライバを利用可能*
- ▶ Ruby版コンフィギュレータの適用

拡張パッケージでサポート(開発中, 未公開)

- ▶ オーバランハンドラ機能
- ▶ 動的生成機能

手動メモリ配置と自動メモリ配置

メモリ配置とは？

- ▶ コードやデータを, どの番地に配置するか？

手動メモリ配置 (HRPカーネルで採用した方式)

- ▶ メモリ配置はユーザが決定する
- ▶ ユーザは, コードやデータが決定したメモリ配置通りに配置されるように, リンクスクリプトを作成する
- ▶ ユーザは, コードやデータが適切に保護されるように, システムコンフィギュレーションファイルを作成する
 - ▶ 静的API (ATT_MEM) を使用して, 先頭番地とサイズで指定したメモリ領域に対して, どのように保護するかを設定する
 - ▶ タスクのユーザスタック領域の自動割付けは使用できず, 先頭番地とサイズで指定することが必要

自動メモリ配置 (HRP2カーネルで採用した方式)

- ▶ ユーザは, コードやデータをどのように保護したいかを, システムコンフィギュレーションファイルに記述する
- ▶ メモリ配置はコンフィギュレータが決定し, リンカスクリプトもコンフィギュレータが生成する

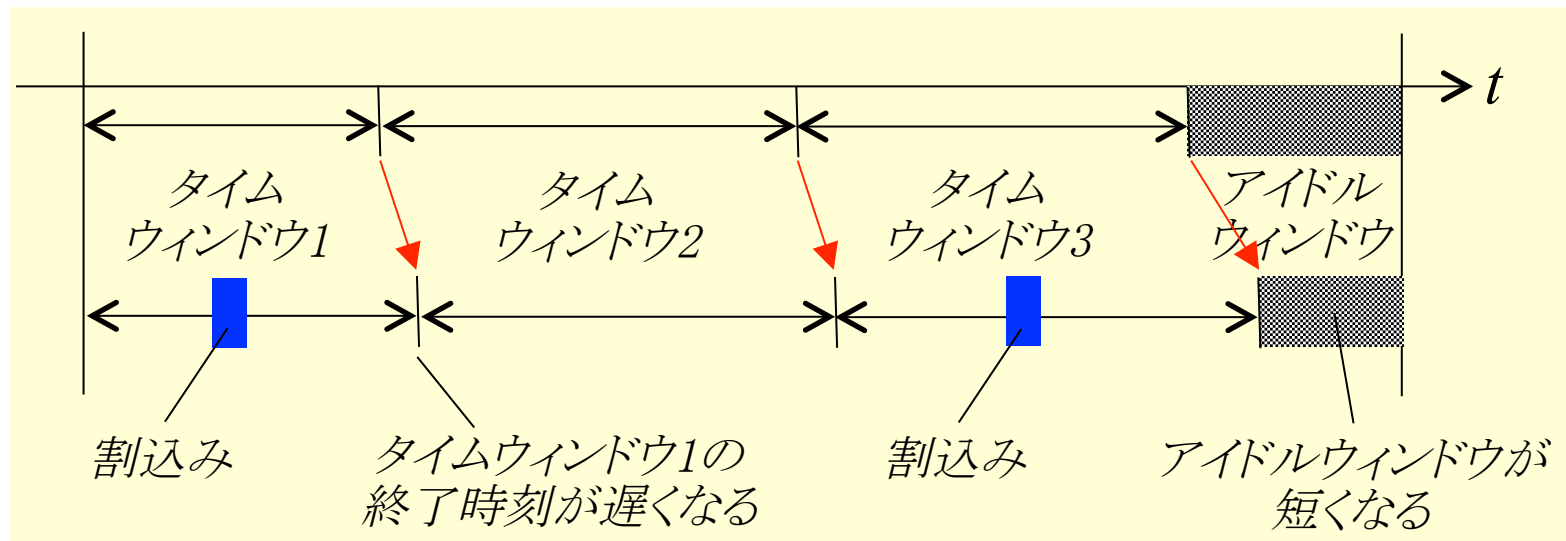
どのように保護したいかの記述方法

- ▶ オブジェクトモジュール毎に保護方法を記述
 - ▶ 静的API(ATT_MOD)を使用して, 指定したオブジェクトモジュールに含まれるコードやデータを, どのように保護するかを設定する
- ▶ セクション毎に保護方法を記述
 - ▶ 静的API(ATT_SEC)を使用して, 指定したセクションを, どのように保護するかを設定する

時間パーティショニング機能の概要

TOPPERS時間パーティショニングスキーム

- ▶ 各パーティションは、システム周期内で各パーティションを実行するタイムウィンドウを決める方式(航空機向け規格であるARINC 653で採用)をベースに、タイムウィンドウによらずに受け付けられる割込みを許すように拡張
- ▶ パーティション内で複数のタスクを実行する場合には、従来通りの方式でスケジュール(階層型スケジュール)



TOPPERS時間パーティショニングスキームの適用

- ▶ 各ユーザドメインをパーティションと扱う. ユーザドメインには複数のタイムウィンドウを割り当てることができる
- ▶ カーネルドメイン(割込み処理はカーネルドメインに属する)は, タイムウィンドウによらずに実行する
- ▶ タイムウィンドウを割り当てられていないユーザドメインは, アイドル時間(タイムウィンドウ内の空き時間, アイドルウィンドウ内の空き時間)に実行する

システム動作モード

- ▶ 保護ドメインをどのように繰り返し実行するかを決定するモード
- ▶ 複数のシステム動作モードを静的に設定したパターンで繰り返すことができる(大きい周期を設定できる)
- ▶ サービスコール(chg_som)により, 指定したシステム動作モードに変更できる

TECSの活用

TECS (TOPPERS組込みコンポーネントシステム)とは？

- ▶ 各種のソフトウェアモジュールを部品化し、必要な部品を組み合わせることによって大規模な組込みソフトウェアを効率的に構築するための技術
- ▶ マルチコアプロセッサシステム/分散システムにおける遠隔呼出し(RPC; Remote Procedure Call)にも対応

ソフトウェアの部品化は必然的な流れ

- ▶ ソフトウェアの再利用性向上のために、ソフトウェア部品化の推進が必要
- ▶ AUTOSARの枠組みでは、アプリケーションはコンポーネントベースで構築する
 - ▶ TECSと似ている面が多い(静的生成や分散システムへの対応など)が、異なる面もある(アプリケーションのみに適用する点)

ASP3, HRP3カーネルへのTECSの導入

システムサービスの構築に利用

- ▶ 「プラットフォームへの適用」
- ▶ システムログ機能, シリアルインタフェースドライバ, システムログタスク, 実行時間分布集計サービスなどをTECSのコンポーネントとして実現
 - ▶ ASP3カーネルとHRP3カーネルでコードを共用できる
- ▶ TECSを使用しないアプリケーション(非TECSアプリケーション)からは従来通りのインタフェースで呼び出せるよう, アダプタモジュール(これもTECSコンポーネント)を用意

カーネル機能のコンポーネント化

- ▶ 「システム全体への適用」
- ▶ タスク, セマフォ, イベントフラグなどが, TECSのコンポーネントとして扱える

ASP3カーネルへのTECS導入で改善できたこと

- ▶ コンポーネントのパラメータが、インスタンス化時の属性の指定で明示可能
 - ▶ ソースコード中で記述されていたパラメータを、コンポーネント記述ファイル(組み上げ記述)の中で記述するようになった
 - ▶ マルチインスタンスの場合、この利点が顕著(ASP3カーネルでは、シリアルインタフェースドライバがこれに該当)
- ▶ 使用する下位レイヤの変更が容易
 - ▶ システムログ機能の例では、異なる低レベル出力処理への変更が容易
- ▶ マルチインスタンス化が容易
 - ▶ システムログ機能の例では、singleton指定を削除するだけで、マルチインスタンス化できる

HRP3カーネルにTECSを導入するメリット

- ▶ 拡張サービスコールのグルーロジックの生成
 - ▶ ユーザタスクから、特権モードで動作するサービス呼び出す場合に、拡張サービスコールを用いる。その際に、グルーロジックが必要(次スライドに例を掲載)
 - ▶ TECSによりグルーロジックが生成できるため、記述が大幅に楽になることに加えて、システムサービスのコードをASP3カーネルと共用できる
- ▶ システムサービスを呼び出せる保護ドメインの制限
 - ▶ システムサービスを呼び出せる保護ドメインを、コンポーネント記述ファイル(組み上げ記述)で記述できる
 - ▶ TECSにより、アクセス権違反のエラーチェックコードを生成できる

グルーロジックの例

- ▶ ユーザタスクから呼び出すためのグルーロジック

```
Inline ER
syslog_wri_log(uint_t prio, const SYSLOG *p_syslog)
{
    return((ER) cal_svc(TFN_SYSLOG_WRI_LOG, (intptr_t) prio,
                        (intptr_t) p_syslog, 0, 0, 0));
}
```

- ▶ `cal_svc`(拡張サービスコールを呼ぶサービスコール)は、
渡すパラメータ数を5つに固定している
- ▶ 拡張サービスコールを登録する静的API

```
DEF_SVC(TFN_SYSLOG_WRI_LOG, { TA_NULL, extsvc_syslog_wri_log,
                               SSZ_SYSLOG_WRI_LOG });
```

グルーロジックの例(続き)

- ▶ 拡張サービスコールとして登録する関数

```
ER_UINT
extsvc_syslog_wri_log(intptr_t prio, intptr_t p_syslog,
                      intptr_t par3, intptr_t par4,
                      intptr_t par5, ID cdmid)
{
    ER_UINT ercd;

    if (!EXTSVC_PROBE_MEM_READ(p_syslog, SYSLOG)) {
        ercd = E_MACV;
    }
    else {
        ercd = (ER_UINT) _syslog_wri_log((uint_t) prio,
                                         (const SYSLOG *) p_syslog);
    }
    return(ercd);
}
```

呼び出し元の保護ドメインID

これも生成可能！

ユーザタスクから読み出し可能なメモリ領域であるかのチェック

ここで呼び出すのが、サービス本体(拡張サービスコールを使わない場合は、これを直接呼ぶ)

コンポーネント記述の例

▶ システムログ機能のシグニチャ

```
signature sSysLog {  
    ER      write([in] uint_t priority, [in] const SYSLOG *p_syslog);  
    ER_UINT read([out] SYSLOG *p_syslog);  
    ER      mask([in] uint_t logMask, [in] uint_t lowMask);  
    ER      refer([out] T_SYSLOG_RLOG *pk_rlog);  
    ER      flush(void);  
};
```

この記述から、p_syslogがSYSLOG型の値を渡すために使われていることがわかる

Ruby版コンフィギュレータの適用

Ruby版コンフィギュレータの特徴

- ▶ コンフィギュレータ本体と各カーネル用の生成スクリプトの両方をRubyで記述
 - ▶ 第2世代カーネル向けでは、コンフィギュレータ本体はC++で記述され、生成スクリプトは専用言語(テンプレートファイル)で記述していた

Rubyスクリプトのサイズ(コメントや空行を含む)

- ▶ コンフィギュレータ本体: 約2400行(静的APIのパーサを含む. 第2世代カーネル向けよりかなり小さい)
- ▶ ASP3カーネルの生成スクリプト: 約2500行
- ▶ HRP3カーネルの生成スクリプト: 約7400行

スクリプト言語をRubyに統一

- ▶ TECSジェネレータ, コンフィギュレータ, その他のユーティリティ群(configure等)を, すべてRubyスクリプトに

今後の開発計画

TOPPERS/ASP3カーネル, HRP3カーネル

- ▶ 対応ターゲットを増やしていく
- ▶ 設計書などのドキュメントの充実を継続的に実施

開発中のカーネル

- ▶ TOPPERS/FMP3カーネル
- ▶ TOPPERS/SSP3カーネル

開発予定のカーネル

- ▶ TOPPERS/HRMP3カーネル
 - ※ HRMP = High Reliable Multiprocessing Profile
 - ▶ 保護機能を持つマルチプロセッサ対応のカーネル
 - ▶ ユーザから要望があり、開発を予定している

TOPPERSプロジェクトの方向性

組込みシステムの開発効率化と品質確保のために

すぐにできること

- ▶ ソフトウェア開発プロセスの地道な改善
- ▶ 人材育成 … 時間はかかるが着手はすぐにできる

中期的に取り組むべきこと

- ▶ 設計抽象度を上げる&設計の早い段階での検証
 - ▶ モデルベース／モデル駆動開発の流れ
 - ▶ シミュレーション(仮想環境)による検証
- ▶ 設計資産の再利用を促進する仕組みの構築
 - ▶ ソフトウェアの再利用は, 差分開発が中心の組込みシステム開発では特に有効
- ▶ 応用分野毎のプラットフォームの構築・活用と共通化
 - ▶ プラットフォーム＝ハードウェア＋OS＋ミドルウェア
- ▶ 非機能要求のモデリングとアシュアランスケース

組込みソフトウェア開発効率化のためのTOPPERS

応用分野毎のプラットフォームの構築・活用と共通化

- ▶ 組込みソフトウェアプラットフォーム構築のための、品質が高く使いやすい(… 使うことで開発の効率化ができる)部品と構築技術を提供する
- ▶ いくつかの分野向けに、プラットフォーム全体を提供する

設計資産の再利用を促進する仕組みの構築

- ▶ ソフトウェアコンポーネント技術(TECS)
- ▶ TOPPERSソフトウェア自身の再利用(業界レベルでの設計資産の蓄積)

人材育成

- ▶ オープンな教材(セミナー資料とソフトウェア)
- ▶ “お手本としてのTOPPERS”

おわりに ～ 恒例のお願い

TOPPERSの活動へのご意見を募集中

- ▶ IoT時代における組込みシステムに必要な技術は？

TOPPERS第3世代カーネルへのご意見を募集中

- ▶ 第3世代へ移行するタイミングは、カーネル仕様を見直すチャンス(かなりフィックスしてきた)

利用事例の報告に関するお願い

- ▶ 利用事例を紹介することは、さらなる採用の促進やプロジェクトの発展につながる → ユーザにも利益に
- ▶ *TOPPERSのユーザには、利用報告をお願いしたい*

会員増加への協力のお願い

- ▶ *TOPPERSのユーザやサポータには、プロジェクトへの入会をお願いしたい*

もう1つ宣伝

TOPPERS/ASP3カーネルについて詳しく知りたい方は...

- ▶ NCES人材育成プログラム(NEP)／enPiT-Pro-Emb
「リアルタイムOSの内部構造」
 - ▶ 日付:2018年8月7日(土), 14日(土)
 - ▶ 時間:9時30分～17時00分(両日とも)
 - ▶ 本会場:名古屋大学 東山キャンパス(満員により締切)
 - ▶ 東京会場:名古屋大学 東京オフィス(受講受付中)
 - ▶ 講師:高田広章, 本田晋也(名古屋大学)
 - ▶ 受講料:4万円

👉 <https://www.nces.i.nagoya-u.ac.jp/NEP/courses/H30/T01.html>

検索: NEP リアルタイムOSの内部構造 2018