

TOPPERSプロジェクトの現状と方向性 + 隔離と共有の難しい関係

2021年6月4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 モビリティ社会研究所 所長・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活動指針(復習)

- ▶ TOPPERSプロジェクトとは？
- ▶ これまでの開発成果, 重点的に取り組んでいるテーマ

TOPPERSプロジェクトの活動状況と成果

- ▶ この1年にリリースした開発成果
- ▶ WGの活動と主な成果, 普及・広報活動の概要

隔離と共有の難しい関係

- ▶ アプリケーション統合の流れ
- ▶ 隔離とその有用性, 隔離の実現
- ▶ 周辺デバイスの共有

おわりに

- ▶ TOPPERSプロジェクトの活かし方
- ▶ 恒例のお願い

TOPPERSプロジェクトの 概要と活動指針（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

- ▶ 高品質な組込みシステムの効率的な開発を支援

組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

主な開発成果物

一般公開しているもの

第1世代カーネル

- ▶ TOPPERS/JSPカーネル, TOPPERS/FI4カーネル
- ▶ TOPPERS/ATK1 (Automotiveカーネル バージョン1)
- ▶ TOPPERS/FDMPカーネル, TOPPERS/HRPカーネル

新世代(第2世代)カーネル

- ▶ TOPPERS/ASPカーネル, TOPPERS/SSPカーネル
- ▶ TOPPERS/FMPカーネル, TOPPERS/HRP2カーネル

第3世代カーネル(ITRON系)

- ▶ TOPPERS/ASP3カーネル, TOPPERS/HRP3カーネル
- ▶ TOPPERS/FMP3カーネル, TOPPERS/HRMP3カーネル

AUTOSAR関連

- ▶ TOPPERS/ATK2 (Automotiveカーネル バージョン2)
- ▶ TOPPERS/A-COMSTACK, TOPPERS/A-WDGSTACK
- ▶ TOPPERS/A-RTEGEN

ミドルウェア

- ▶ TINET (TCP/IPスタック), FatFs for TOPPERS
- ▶ TOPPERS/ECNL (ECHONET Lite通信ミドルウェア)
- ▶ RLL (Remote Link Loader), DLM (Dynamic Loading Manager)

ツール, ソフトウェアプラットフォーム, その他

- ▶ TECS (TOPPERS組込みコンポーネントシステム)
- ▶ TOPPERS BASE PLATFORM (ST, CV, RV)
- ▶ SafeG (デュアルOSモニタ), SafeG64, SafeG-M
- ▶ EV3RT (LEGO Mindstorms EV3向けSPF)
- ▶ MDCOM (MultiDomain Communication Module)

教育コンテンツ

- ▶ 初級・中級実装セミナー教材
- ▶ 基礎1・基礎2・基礎3実装セミナー教材
- ▶ 基礎ハードウェア設計セミナー教材
- ▶ ETロボコン向けTOPPERS活用セミナー教材

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 **改訂時期に来ている**

- ▶ 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仕様ベース)

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)

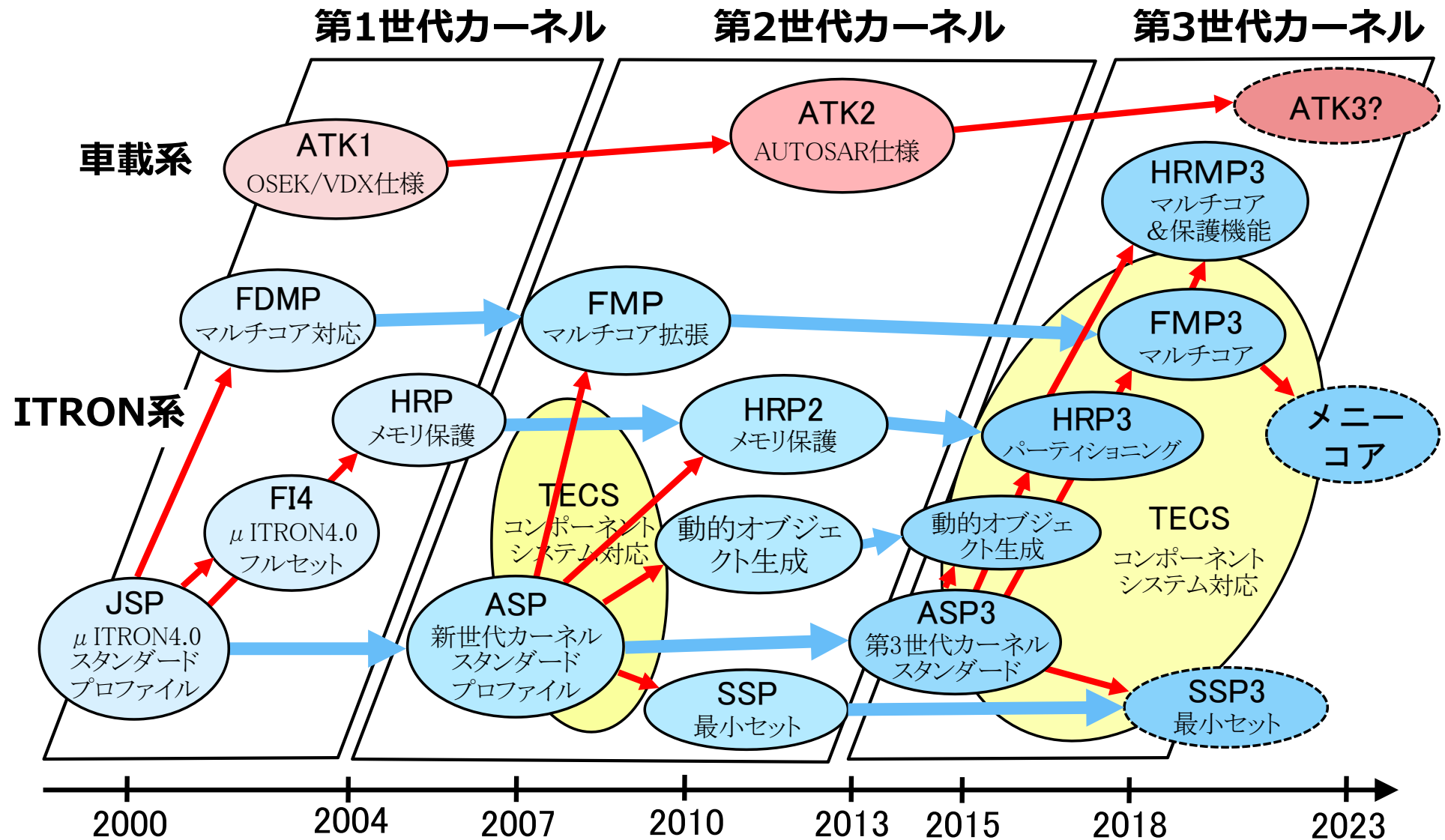
組込みシステム向けSPFと開発支援ツール

- ▶ TOPPERS BASE PLATFORM, ホームネットワーク
- ▶ シミュレーション環境(箱庭), 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), ドメイン間通信(MDCOM)

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第3世代カーネルの開発状況

ほぼ完成, 改良・保守フェーズに

- ▶ ASP3: スタンダードプロファイル
- ▶ HRP3: 保護機能 (メモリ保護, 時間パーティショニング)
- ▶ FMP3: マルチコア
- ▶ HRMP3: 保護機能 + マルチコア

TOPPERS/FMP3カーネル, HRMP3カーネルのTECS対応

- ▶ TECS側の開発が進捗 (早期リリース段階), もう少し?

開発中のカーネル

▶ TOPPERS/SSP3カーネル ... 早期リリース中 メニーコアプロセッサ向けのカーネル (名称未定)

- ▶ FMPカーネルはメニーコアプロセッサにも有用なことが示せているが, FMP3カーネルでは対象から除外

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

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”

新しい技術の導入

- ▶ 新しい技術を導入した開発

TOPPERSプロジェクトの 活動状況と成果

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

一般公開(2020年6月～2020年11月)

- 7月15日 MDCOM Release 1.2.0
- 7月25日 SafeG64 バージョン1.1.0
- 8月18日 SafeG-M バージョン1.0.0(最初の一般公開)
- 8月20日 Athrill rh850版(最初の一般公開)
- 11月6日 TOPPERS第3世代カーネル統合仕様書 Release 3.4.2 **LTS**
- 11月6日 TOPPERS/HRMP3カーネル Release 3.1.2 **LTS**
- 11月21日 TOPPERS/ASP RISC-V向け個別パッケージ Release 1.9.4

一般公開(2020年12月～2021年5月)

12月23日 TOPPERS第3世代カーネル統合仕様書 Release 3.5.0

12月23日 TOPPERS/ASP3 Release 3.6.0

12月23日 TOPPERS/HRP3 Release 3.3.0

12月23日 TOPPERS/HRMP3 Release 3.2.0

1月8日 TOPPERS/FMP3 Release 3.2.0

5月26日 箱庭プロトタイプモデルA:単体ロボット向けシミュレータ

5月29日 TOPPERS BASE PLATFORM(ST/RV) V1.4.2
TOPPERS BASE PLATFORM(CV) V1.1.2

早期リリース(2020年6月～2021年5月)

9月25日 TOPPERS BASE PLATFORM(RV) V0.1.3

12月23日 TOPPERS BASE PLATFORM(RV) V0.1.4

今回リリースした開発成果

一般公開

6月4日 SafeG64 バージョン1.2.0

早期リリース

6月4日 TECSジェネレータ 1.8RC9

WGの活動と主な成果

教育WG

- ▶ 新基礎3セミナー, 上級1セミナー
- ▶ TOPPERS BASE PLATFORM (ST, CV, RV) の開発
- ▶ 教育WGシールドの開発
- ▶ IoTデバイス (BLEリモコンプラモデル戦車) の開発
- ▶ 教育セミナーの実施 **昨年度は実施できず**

TECS WG

- ▶ TECSジェネレータ+プラグイン (継続的に拡張・改良)
 - ▶ TOPPERS/HRMP3, FMP3対応
- ▶ TECSFlow, TECSUnit, mruby on EV3RT+TECS
- ▶ 各種のデバイスドライバ開発
- ▶ TECS2MrubyBridgePlugin
- ▶ ECHONET Lite との連携

ホームネットワークWG

- ▶ Azure IoT接続のWi-Fi対応
- ▶ TINETの脆弱性調査検討

箱庭WG

- ▶ 単体ロボット向けシミュレータ(試行版の公開)
- ▶ ROS・マルチECU向け, ロボット間協調動作向けシミュレータ
- ▶ アセット間の時間同期機構の技術検討
- ▶ IoTシステム開発効率化への試み
- ▶ 教育機会での箱庭成果物の採用

海外展開WG

- ▶ TOPPERS第3世代カーネル統合仕様書の英語化

普及・広報活動の概要

カンファレンス実行委員会

- ▶ TOPPERSカンファレンス

展示会運営委員会 昨年度はET/IoT (オンライン) のみ参加

- ▶ ET/IoT, ET/IoT-West, ESEC, ARMシンポジウム

TOPPERS開発者会議実行委員会

- ▶ TOPPERS開発者会議
- ▶ TOPPERS活用アイデア・アプリケーション開発コンテスト
- ▶ オープンソースカンファレンス(OSC)

広報戦略TF, エコシステム

- ▶ 外部(CQ出版等)との連携によるセミナー開催
- ▶ ウェブサイトの見直し

TOPPERSプロジェクトニュースレター

その他の活動

技術報告会議・技術検討会議

7月16日 ESP32をWIFIモジュール化

8月27日 SafeG/SafeG64/SafeG-M: ARMプロセッサ向けの仮想化ソフトウェア

ETロボコンへの協力

- ▶ ETロボコンに特別協賛
- ▶ ソフトウェアプラットフォームとシミュレーション環境の提供
 - ▶ TOPPERS/EV3RT
 - ▶ 箱庭
- ▶ ETロボコン向け TOPPERS活用セミナー
- ▶ TOPPERS賞の授与
 - ▶ 実装技術の優れたチームを表彰

GitHubの活用

- ▶ GitHubにTOPPERSプロジェクトのOrganizationアカウントを取得し, WGの活動に使用
- ▶ TOPPERS Contributed Software on GitHub

Qiita Advent Calendar 2020

- ▶ 2020年12月1日～25日の25本の記事を掲載



アプリケーション統合の流れ

従来：専用システム

- ▶ 高い信頼性・安全性が求められる組込みシステムは、1つのアプリケーション専用の開発されることが多かった
 - ▶ 過度な複雑性を持ち込みたくないため
- ▶ その結果、複雑な機器（自動車、航空機など）では、組込みシステム（コンピュータ）が増え過ぎることが課題に
 - ▶ 1台の自動車に100個ものECU（コンピュータ）

アプリケーション統合（ECU統合）

- ▶ 複数の組込みシステム（コンピュータ）を1つに統合する
 - ▶ 自動車分野では、「ビークルコンピュータ」、「セントラルECU」などと呼ばれている
- ▶ 1つの故障／不具合が、広範囲に影響を及ぼすように
 - ▶ 隔離（isolation）が重要に

隔離とその有用性

隔離 (isolation) とは？

- ▶ [ざっくりした定義] アプリケーション (システム, コンポーネント) 間で影響 (特に悪影響) が波及しないようにすること
- ▶ 保護, FFI (Freedom From Interference) とも呼ばれる

隔離の有用性

- ▶ 安全性 (safety) の向上
 - ▶ あるアプリケーションの故障で, システム全体が致命的な障害を起こさない
- ▶ セキュリティの確保・向上
 - ▶ ある脆弱性 (security) の悪影響を最小限にする
- ▶ ソフトウェア開発 (工数) の最適化
 - ▶ アプリケーション毎に, それに要求される信頼性が満たせるように開発すれば良い

車載組込みシステムにおける事例(1)

- ▶ 2009年～2010年のトヨタ車の急加速問題
 - ▶ 多くのユーザが, トヨタ車を運転中に急加速が発生し, ブレーキも効かなかったと主張
 - ▶ エンジン(厳密には電子スロットル)の電子制御装置の不具合が疑われた
- ▶ 2011年2月に発表されたNHTSA(米運輸省高速道路交通安全局)とNASAによる最終報告では, 急加速が起こる原因は, 電子制御装置には見つからなかったとされた
- ▶ エンジン制御システムが, ブレーキ制御システムと独立していることが, 重要な論拠となった
 - ▶ 独立している2つのシステムが, 同時に誤動作する確率は, 片方のシステムが誤動作する確率に比べて, 極めて低い

車載組込みシステムにおける事例(2)

- ▶ テスラ社のメディアコントロールユニット(MCU)の欠陥で、タッチスクリーンの操作が不可能になる問題
 - ▶ NHTSAによる分析で、MCUのストレージ(eMMC)が、書換え回数の上限を超えることが原因と判明
 - ▶ 2021年2月に、13.5万台のリコールが発表された
- ▶ タッチスクリーンの操作が不可能になると...
 - ▶ 後方カメラ, 曇りとり, 霜とりなどの機能が操作不能に
 - ▶ 「オートパイロット」を起動したときにドライバに音で知らせるチャイムを止めてしまう可能性
 - 安全性に影響があると判断された
- ▶ 直接的には、ストレージの書換え回数上限の考慮ミスだが、多くの機能をMCUに統合しすぎたのも一因と思える

隔離の実現

影響が波及する原因

- ▶ システム／コンポーネント間の影響の波及は、共有するリソースを通じて起こる
 - ▶ 逆に言うと、リソースを全く共有していなければ(つまり、アプリケーション統合がされていないならば)、影響は波及しない

アプリケーション統合における共有リソース

- ▶ ハードウェアリソース
 - ▶ プロセッサ
 - ▶ メモリ
 - ▶ バス(インターコネクト)
 - ▶ 周辺デバイス
- ▶ ソフトウェアリソース

共有リソースの分離

- ▶ 共有リソースは、空間分割または時分割により、各アプリケーションに共有されている
- ▶ 各アプリケーションが共有リソースを使う空間／時間を分離(partitioning)することで、隔離が実現できる

空間のパーティショニング

- ▶ 典型例) メモリ保護(メモリ空間のパーティショニング)
- ▶ メモリ空間は(おおよそ)一様なので、パーティショニングは容易であり、実現技術も確立している

時間のパーティショニング

- ▶ 典型例) プロセッサの時間保護(プロセッサ時間のパーティショニング)
- ▶ 時間は一様でないので、悩ましい...
 - ▶ TOPPERSでは、(一応の)解決法を用意

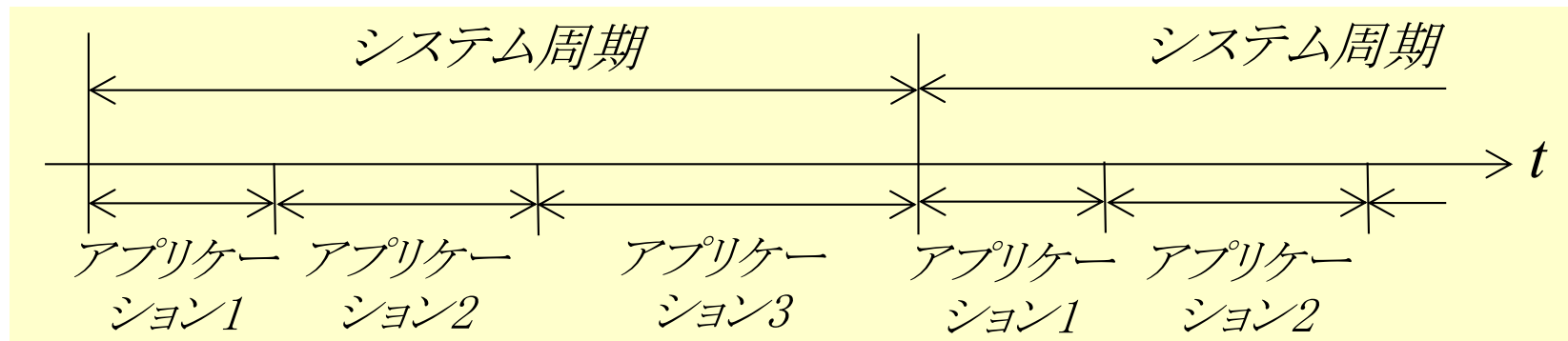
ARINC-653の時間パーティショニング

▶ ARINC-653とは？

- ▶ 航空機向けのOSのインタフェース仕様の標準
- ▶ アプリケーション統合 (IMA) が主なターゲット

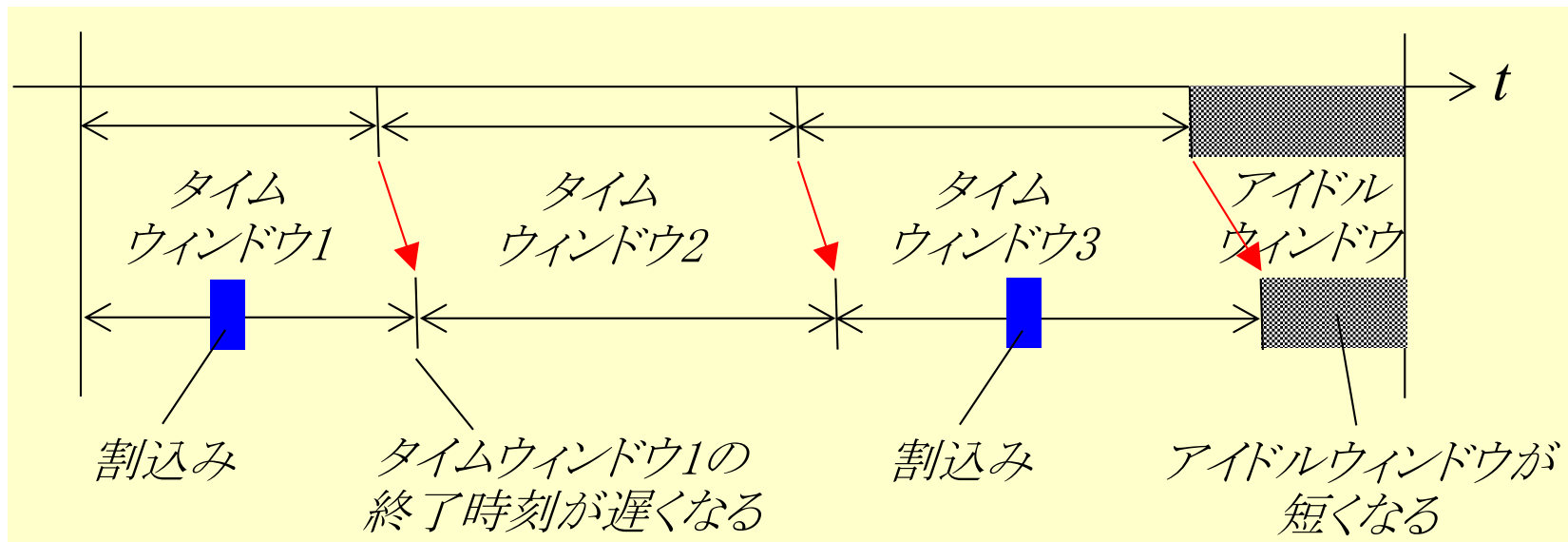
▶ スケジューリング方式の概要

- ▶ 静的に決定したシステム周期を複数のタイムウィンドウに分割し、アプリケーションを、それに割り付けられたタイムウィンドウ内で実行する
- ▶ アプリケーションに対する割込みも、アプリケーションに割り当てられたタイムウィンドウ内でのみ受け付けられる



TOPPERS時間パーティショニングスキーム

- ▶ ARINC 653の方式をベースに, タイムウィンドウによらずに実行される処理(典型的には, 割込み)を許すように拡張
- ▶ システム周期の最後に, アイドルウィンドウ(どのアプリケーションにも割り当てられていないタイムウィンドウ)を置く
 - ▶ タイムウィンドウによらずに実行される処理が使った時間は, アイドルウィンドウで吸収
- ▶ ATK2カーネル(ATK2-SC3-TP), HRP3カーネルに実装



隔離の実現形態

- ▶ OSの保護機能による隔離
 - ▶ メモリ保護機能やプロセッサ時間保護機能により, アプリケーション間を分離
- ▶ ハイパバイザ(仮想マシン)による隔離
 - ▶ 1つのコンピュータ上で複数の仮想マシンを動作させる
 - ▶ 仮想化を支援するハードウェア機能(制御システム向けマイコンにも, 持つものが登場)により, 小さいオーバヘッドで実現可能に
- ▶ コンテナによる隔離(OSによる隔離の一種)
 - ▶ 1つのOS上に複数の実行環境を構築する
 - ▶ (現時点では)UNIX系OS上の技術
- ▶ マルチコアによる隔離
 - ▶ プロセッサを共有するのをやめる

残る共有リソース

- ▶ プロセッサとメモリの分離は実現できたが、共有リソースは他にも色々ある

(まずは)周辺デバイス

- ▶ ネットワーク(およびコントローラ)
 - ▶ ストレージデバイス(およびコントローラ)
 - ▶ HSM(ハードウェアセキュリティモジュール)
 - ▶ GPU, AIエンジン, その他のアクセラレータ
- 一般論はあるものの、デバイス毎に性質が異なり、個別に検討することも必要

さらに...

- ▶ バス(インターコネクト) … マルチコアで問題に
- ▶ キャッシュ(容量, 帯域), メモリ帯域
- ▶ ソフトウェアリソース(ソフトウェアアーキテクチャに依存)

周辺デバイスの共有

基本構成

- ▶ 共有する周辺デバイスのデバイスドライバは、どこかに1つ必要(これを共有デバドラと呼ぶ)
 - ▶ ハイパバイザによる隔離の場合は、仮想マシン内に仮想デバイスドライバも必要
- ▶ 周辺デバイスを扱うミドルウェアは、1つだけ持つ場合と、アプリケーション(パーティション)毎に持つ方法がある

共有デバドラの実現形態

- ▶ 方法1: 共有デバドラは、アプリケーションから特権呼出し／ハイパコールで呼び出す
- ▶ 方法2: 共有デバドラをサーバタスク(処理)として実現し、アプリケーションからタスク(処理)間通信で呼び出す

2つの実現形態の比較

- ▶ 方式1の欠点: 共通デバドラは高い特権レベルで実行されるため、そこに不具合や脆弱性があると、システム全体に影響する
- ▶ 方式2の欠点: オーバヘッドが大きい

共有デバドラによる隔離の実現

- ! ストレージのリード／ライト処理を例にすると考えやすい
- ▶ 周辺デバイスを空間分割する場合、共有デバドラがアクセス場所が正当かのチェックを行う
- ▶ 周辺デバイスの処理に時間がかかる場合、共有デバドラは、各アプリケーションから依頼された処理量を管理し、各アプリケーションに割り当てられた処理量を超えないように制御する

実現形態とプロセッサの時間パーティショニングとの関係

- ▶ 方法1: 共有デバドラは, それを呼び出したアプリケーションに割り当てられた時間内で実行する
- ▶ 方法2: 共有デバドラは, それ用のパーティション内で実行する(共有デバドラにプロセッサ時間を割り当てる)

方式1の課題

- ▶ 共有デバドラがクリティカルセクションを実行中に, アプリケーションに割り当てられた時間がなくなったら?
 - ▶ アプリケーションを切り換えると, 他のアプリケーションが共通デバドラを呼び出した場合に処理できない
 - ✓ 一種のlock holder preemption
 - ▶ アプリケーション切換えを遅延させると, プロセッサの時間パーティショニングの正確度が落ちる
 - ✓ TOPPERS時間パーティショニングスキームでは, 切換え遅延時間を, アイドルウィンドウで吸収することは可能だが...

方式2の課題

- ▶ あるアプリケーションが共有デバドラに負荷を掛け(例えば, 多数回呼び出し), 他のアプリケーションからの呼び出しを処理する時間がなくなる可能性
 - ▶ つまり, 「共有デバドラの処理時間」という共有リソースを通じた影響の波及が課題に

今後の取り組み計画

- ▶ デバイスドライバの設計ガイドラインとサンプルを作成する

お断り (注意事項)

共通原因故障の考慮が必要

- ▶ 今日のプレゼンでは言及しなかったが、何かを「共有」する際には、共通原因故障についても考慮することが必要
 - ▶ (当然ながら)1つのコンピュータに複数のアプリケーションを統合すると、コンピュータのハードウェアの故障で、複数のアプリケーションが動作しなくなる
 - ▶ 「2つの実現形態の比較」の「方式1の欠点」は共通原因故障の問題と見ることもできる

おわりに

TOPPERSプロジェクトの活かし方

- ▶ TOPPERSの開発成果物を利用するだけが、TOPPERSプロジェクトの活かし方ではない

組込みソフトウェア開発企業には

- ▶ オープンソース開発によるオープンイノベーション
 - ▶ 組込みシステムのソフトウェアプラットフォームを整備したいが、自社単独では難しく、他社と協力したい
 - ▶ 新しい開発技術を評価するにあたり、他社と協力したい
- ▶ 自社のソフトウェア／技術(の一部)を公開(オープンソース化)して、周知度を上げたい

個人や大学のオープンソース開発者には

- ▶ 開発したオープンソースソフトウェアを世に問いたい
- ▶ 一緒に開発する仲間を募りたい

おわりに ～ 恒例のお願い

TOPPERSへの入会のお願い

- ▶ *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

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

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世代カーネルに入れるべき機能は？