TOPPERSプロジェクトの方向性と取り組む予定のテーマ

2012年6月12日

高田 広章

名古屋大学 大学院情報科学研究科 教授
附属組込みシステム研究センター長

NPO法人 TOPPERSプロジェクト 会長

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

目次

TOPPERSプロジェクトの現状と方向性

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　ほぼ完了

- 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

重点的に取り組んでいるテーマ

次世代のリアルタイムカーネル技術

- ! 高信頼性・安全性・リアルタイム性を追求
- TOPPERS新世代カーネル（ITRON仕様からの発展）
- 次世代車載システム向けRTOS（AUTOSAR仕様をベース）

ソフトウェア部品化技術，セキュリティ向上技術

- TECS（TOPPERS組込みコンポーネントシステム）
- SafeG（高信頼組込みシステム向けデュアルOSモニタ）

組込みシステム向けプラットフォームと開発支援ツール

- 開発支援ツール（シミュレータ，可視化ツール）
- 宇宙機向けソフトウェアプラットフォーム（SpaceWire OS）

技術者育成のための教材開発

- プラットフォーム技術者育成のための教材
- ETロボコン向けプラットフォームと教材の提供

TOPPERS新世代カーネルの必要性

μITRON4.0仕様の公表から，すでに10年以上が経過

組込みシステムにおける要求の変化

- システム/ソフトウェアの一層の大規模化・複雑化
- これまで以上に高い信頼性・安全性
- 小さい消費エネルギーで高い性能

μITRON4.0仕様以降の各方面の技術開発成果

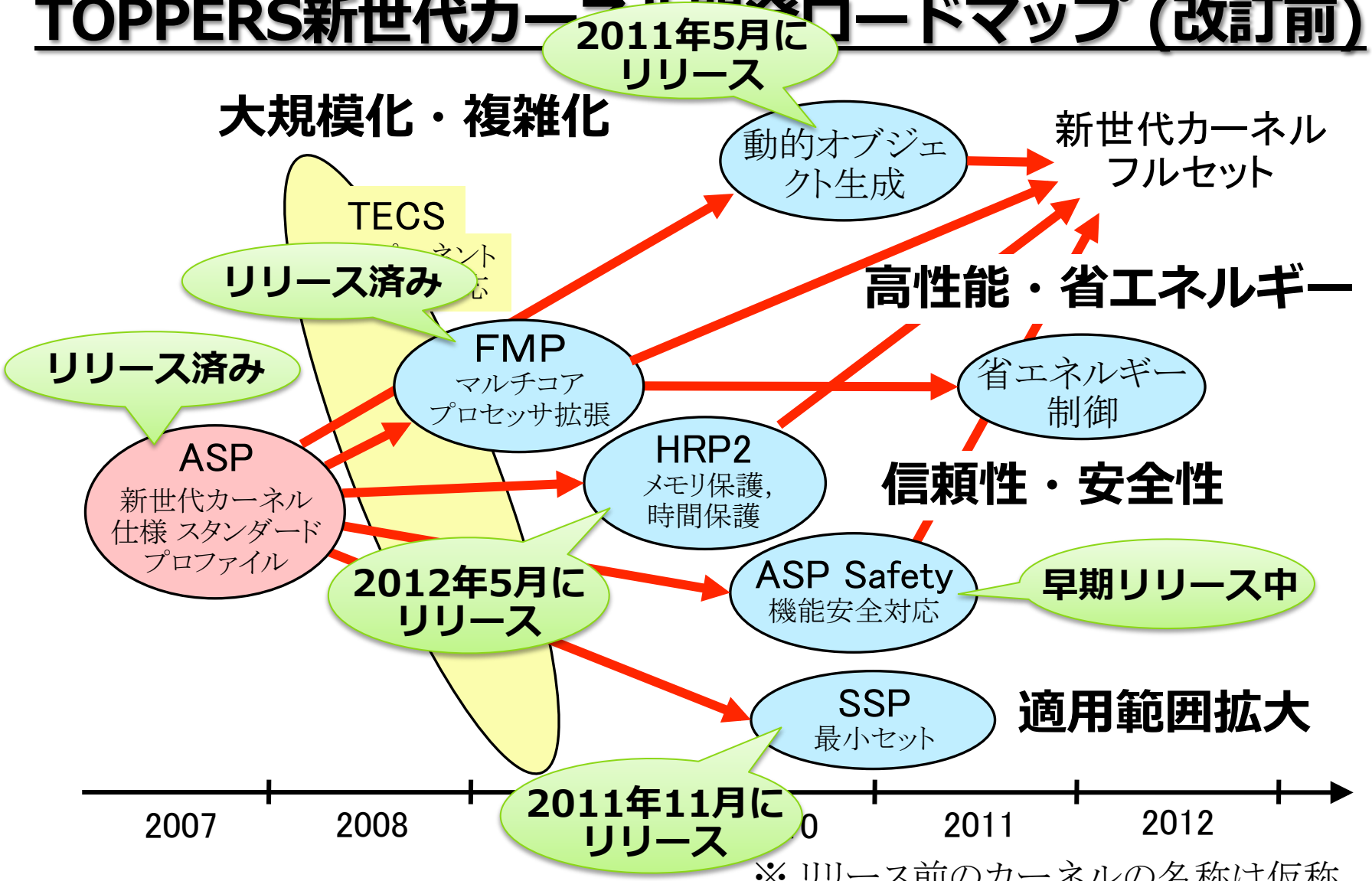
- マルチコアプロセッサ対応
- 保護機能（メモリ保護，時間保護）
- 機能安全対応，省エネルギー制御
- コンポーネントシステム対応

μITRON4.0仕様で完成度が低かった箇所の改良

- システムコンフィギュレーション手順など

➔ これらの要求にこたえる新しいカーネルが必要

TOPPERS新世代カーネル開発ロードマップ (改訂前)



組込みシステムの今後の変化

制御と情報処理の統合（統合システム，融合システム）

- 情報通信技術と組込みシステム技術を活用したスマート社会を構築することが世界的な流れ
  - スマートグリッド，スマートコミュニティ，エネルギーITS，…
- 組込みシステムと情報システムを結合した大規模な統合システム（融合システム）の構築が重要に

ネットワークによる機能再配置 ← *クラウドコンピューティング*

- それぞれのサービスの複雑化はさらに進むと思われる
- すべての機器がネットワーク接続されれば，すべての機器が汎用・多機能である必要はない
  - ! パラダイムチェンジの時期の見極めが難しい

消費電力あたりの性能の向上

- 新しいハードウェア技術の導入が必要

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

保護機能からパーティショニングへ

TOPPERS新世代カーネル開発ロードマップの改訂

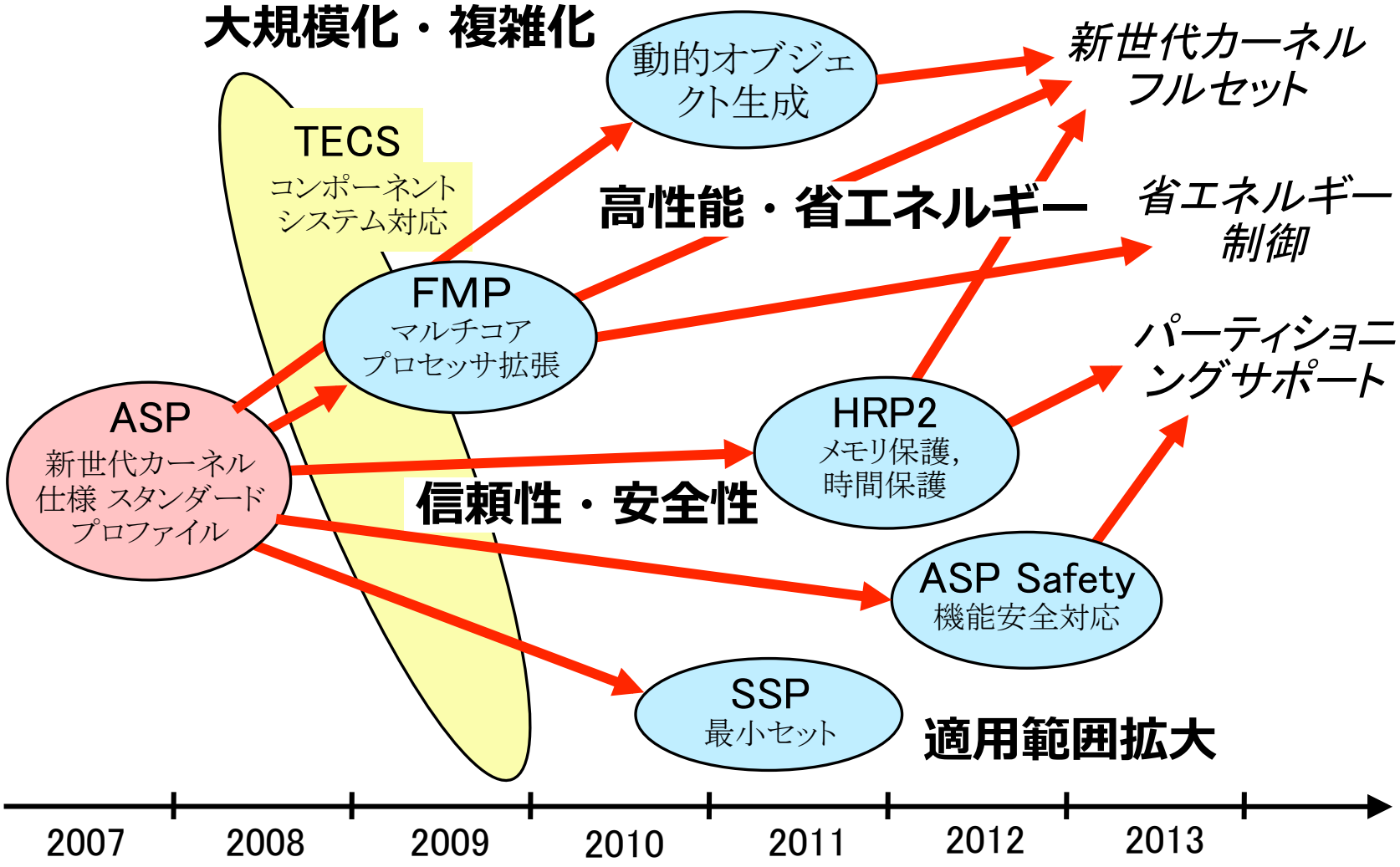
新世代カーネルの開発状況

- これまでの開発ロードマップに掲載されているカーネルの大部分をリリース
  - 「省エネルギー制御」だけが残っている

開発ロードマップの改訂

- リリース済みのカーネルを，リリースした時期に合わせて再配置
- 今後の開発の方向性として次の3つを挙げた
  - 新世代カーネルフルセット
  - 省エネルギー制御
  - パーティショニングサポート … *新たな開発課題*
- 具体的な開発計画は今後検討

TOPPERS新世代カーネル開発ロードマップ (改訂版)



保護機能を導入した目的

OSの保護機能とは？

- OS上で動作するあるアプリケーションが誤動作した場合に，その誤動作がOS自身や他のアプリケーションに障害を引き起こすのを防ぐための機能

μITRON4.0仕様に保護機能拡張を導入した目的（当初）

(1) デバッグ支援のため
- バグの切り分けが容易に

(2) 信頼性向上のため
- システムの一部分に問題があっても，システム全体の動作を継続できる
- 信頼性要件の異なるモジュールが共存できる

(3) セキュリティ確保のため
- 悪意を持ったプログラムをダウンロードした場合に対応

保護機能に対する最近の目的/要求

開発/検証コストの最適化

- 各コンポーネント(モジュール)に要求される信頼性レベルに必要なプロセスで開発/検証すればよい
  - 機能安全規格の安全度水準(IEC 61508のSIL, ISO 26262のASIL)
  - 情報セキュリティ評価基準(ISO/IEC 15408)の評価保証レベル(EAL)
- あるコンポーネントを修正した際に, 他の修正しないコンポーネントの再検証が不要(または簡素化できる)

→ *あるコンポーネントの誤動作が他のコンポーネントに影響しないことの, より厳格な保証が求められる*

“パーティショニング”

パーティション単位の停止/再起動

- あるパーティションでエラーが検出された場合に，パーティション単位で停止または再起動できること
  - 信頼性レベルが低いパーティションのエラーのために，信頼性レベルが高いパーティションが停止するのを避ける

資源共有と保護

- 複数のパーティションで資源（特に，ネットワーク関連の周辺デバイス）を共有する際に，あるパーティションの誤動作が他に波及しないこと

既存のRTOS仕様における保護機能と課題

μITRON4.0/PX（保護機能拡張）仕様の保護機能

- 「保護ドメイン」の単位での空間保護
  - メモリ保護（メモリに対するアクセス保護）
  - カーネルオブジェクトに対するアクセス保護
- タスクの単位での時間保護
  - オーバランハンドラ（タスクの実行時間が設定した上限値を超えた場合に例外処理）
- 特権サービスの呼び出しによる資源共有
  - 拡張サービスコール

μITRON4.0/PX仕様の課題

- パーティションを実現するためには，多くの機能をミドルウェア等で実現する必要があり，実装負荷が大きい

AUTOSAR OS仕様の保護機能

- 「OSアプリケーション」の単位での空間保護
  - メモリ保護（メモリに対するアクセス保護）
  - サービス保護（カーネルオブジェクトに対するアクセス保護機能など）
- タスクと割込み処理（ISR）単位での時間保護
  - タスクとISRの実行時間の上限監視
  - タスクとISRの到着間隔の下限監視
  - 割込み禁止時間とリソースロック時間の上限監視
- 特権サービスの呼び出しによる資源共有
  - 信頼関数
- OSアプリケーション単位の停止/再起動

AUTOSAR OS仕様の時間保護の問題

- 時間保護の単位が小さいこと
  - あるパーティションを構成するタスクや割込み処理が変化する場合（例えば，特定のパーティションのみ再起動したい場合）にうまく対応できない
  - パーティションの単位で時間を保護したい
- 他のパーティションからの影響 … *許容できる問題*
  - あるパーティションが実行されるタイミングが，他のパーティションの動作の影響を受ける
- 実行オーバヘッドが大きい
- さらに，タスク/ISRが信頼関数を呼び出している途中で，タスク/ISRの実行時間が上限に達した場合の扱いに問題がある … *本質的な問題*

ARINC 653（航空機向けRTOS仕様）の保護機能

- 「パーティション」の単位での空間保護
- 「パーティション」の単位での時間保護
  - 固定周期内で，各パーティションを実行する時間が固定されている → *パーティション間での干渉が全くない*
  - パーティション内でタスク（プロセスと呼ばれている）をスケジューリング
- パーティション間通信による資源共有
- パーティション単位での停止/再起動

ARINC 653の時間保護の問題

- 割込み処理が使えないため，応答性が落ちる（逆に言うと，高性能なプロセッサが必要）

策定するパーティショニング機能

パーティション単位での空間保護

- 従来の方法で問題なし

パーティション単位での時間保護

- ARINC 653の方式（固定周期内で各パーティションを実行する時間を決める）方式をベースとする
  - ただし，特権レベルでの割込み処理は使えるようにする
    … *それによるパーティション間の干渉は許容する*
- パーティション内でタスクを優先度ベーススケジューリング

資源共有

- 特権サービス呼び出しは，サポートしないか，サポートするとしても限定的な機能に

パーティション単位での停止/再起動

パーティショニング機能の実装

TOPPERS新世代カーネルへの導入

- TOPPERS新世代カーネルの1つとして，パーティショニング機能を持ったリアルタイムカーネルを開発
  - 各パーティションでTOPPERS/ASPカーネルが動作するイメージ

次世代車載システム向けRTOSへの導入

- 次世代車載システム向けRTOSへの導入を検討する
  - AUTOSAR OS仕様に対する拡張（逸脱）になるため，慎重な検討が必要

外部への提案

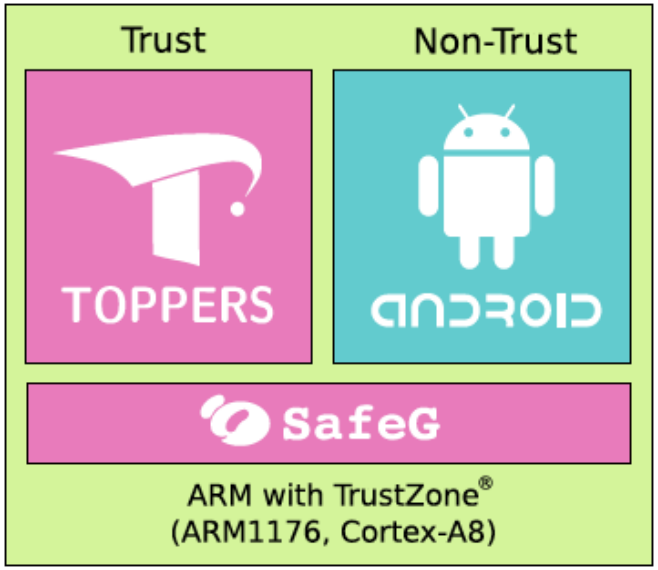
- TOPPERSプロジェクトの外部へも，同じコンセプトのパーティショニング機能の実装を働きかける

SafeGによるセキュリティ向上技術

SafeG：高信頼組込みシステム向けデュアルOSモニタ

SafeGとは？

- 1つのマイクロプロセッサ上で，汎用OSとRTOSを安全に共存して動作させるデュアルOSモニタ
- 名古屋大学 組込みシステム研究センター（NCES）と高田研究室で開発



SafeGの最大の特徴

- ARM TrustZone技術を用い，RTOSをTrust状態，汎用OSをNon-Trust状態で実行し，RTOSと汎用OSを時間的・空間的に分離
- 汎用OSにバグやセキュリティホールがあり，汎用OSのカーネルが誤動作したり，特権モードで不正なプログラムが動作しても，RTOS側を保護できる

開発の背景

- 最近の組込みシステムは，ネットワークに接続することで豊富な機能を提供することが求められる一方で，高い安全性やリアルタイム性が必要な処理も実行する必要
- ネットワーク接続による豊富な機能の実現と高信頼性の確保という相反する要求に，1つのOSだけで応えることは難しい
- LinuxやWindowsのような汎用OSと，μITRON仕様やAUTOSAR仕様のようなRTOSを併用するアプローチが有効
- 仮想マシン（VM）やハイブリッドOSの技術は，実行時オーバヘッドが大きいという問題や，OS自身の不具合には対処できないという問題がある

SafeGのその他の特長

- RTOS側のリアルタイム性を保証
  - RTOS側の割込みが常に優先され，それによりOSの切換えを行うため，汎用OSの実行によって，RTOS上で動作するアプリケーションのリアルタイム性が阻害されない
- 小さい実行時オーバヘッド
  - ARM TrustZone技術を活用することによって，OS間の切換え時間が短い（200サイクル程度）
- 汎用OSに加える修正が少ない
  - 汎用OSのバージョンを上げる必要がある場合にも，容易に追従することができる
- SafeG自身は小規模で信頼性確保が容易
  - SafeG自身は，コードサイズ（バイナリ）が2KB程度の小規模なソフトウェアで，信頼性を確保することが容易

SafeGの開発状況と計画

これまでのリリース

- 2011年1月：SafeG Release 0.1 … 早期リリース
- 2011年6月：SafeG Release 0.2 … 一般公開
  - 汎用OSにLinuxおよびAndroidを，RTOSにTOPPERS/ASPカーネルを使用
- 2011年12月：SafeG Release 0.3 … 一般公開
  - Trust側をOSなしに対応
  - マルチコアプロセッサにテスト対応（RTOSにTOPPERS/FMPカーネルを使用）
  - OS間通信機能をテスト実装
  - 統合スケジューリング機構（汎用OS側の一部を，RTOSの低優先度タスクより優先して実行する）をテスト実装

対応プロセッサとハードウェア機能

- ARM TrustZoneを搭載したプロセッサ
  - ARM Cortex-A9，A8，A5
  - ARM1176
- ハードウェアが備えるべき機能
  - メモリとデバイスを，Trust側とNon-Trust側に分離できること … *この機能がない場合でもSafeGは動作するが，空間保護はされない*
  - Trust側とNon-Trust側の割込みを区別できること

今後の開発計画

- 対応ボードの追加
- 仮想化によるintrospectionモニタ
- マルチコアプロセッサ対応におけるロックホルダプリエンプション問題の解決/軽減

SafeGによるセキュリティ向上

ファイアウォールとしてのSafeG

- 汎用OS側にセキュリティホールがあっても，RTOS側を保護できる
- Safety-criticalな処理はRTOS側で実行する

自己診断の機構としてのSafeG

- RTOS側で，汎用OSに対する侵入検知を行う
  - 汎用OSに侵入できても，侵入検知処理を停止させることができない（二重の防御機構）
  - 侵入を検知したら，安全な状態に移行するなどの対応
- 実際には，侵入検知処理自身は汎用OS側で実行し，RTOS側では，侵入検知処理が正しく動作していることをチェックする

その先の取り組み

リアルタイムカーネルから周辺技術へ

次世代のリアルタイムカーネル技術の開発状況

- TOPPERS新世代カーネルの技術は，残っている課題（省エネルギー制御，パーティショニング）はあるものの，出口が見えつつある
- さらに次の世代のカーネル技術までは少し時間がありそう
  - 次の大きい課題は，メニーコアプロセッサ（GPGPUも含む）や動的再構成ロジックへの対応か？

周辺技術への取り組み状況

- TOPPERSプロジェクトとしては，周辺技術にも力を入れて取り組んでいる
  - TECS，SafeG
  - ミドルウェア，ソフトウェアプラットフォーム
  - 開発支援ツール（シミュレータ，可視化ツール）

Smart Futureに向けての研究開発課題（再掲）

- Safety & Security
- Ecology（高エネルギー効率）
- Connectivity

さらなる展開の候補領域（これに限らない）

- データ処理プラットフォーム
  - 組込みDBMS
  - データストリーム処理
- スクリプト言語
  - 軽量Ruby, Lua
- システムメンテナンスのフレームワーク
- アプリケーション領域
  - スマートグリッド, ITS, ヘルスケア, …

議論と開発への参加のお願い

議論への参加のお願い

- 研究開発課題(上記)から研究開発項目へ落とし込むための議論が必要
  - 技術検討会議，TOPPERS開発者会議，…
  - TOPPERS活用アイデア・アプリケーションコンテスト

(共同)開発者を募集

- TOPPERSの技術開発・ソフトウェア開発に参加して下さる方を常に募集している
  - 大きいテーマは，コンソーシアム型での開発へ