# TUMBLER: 組み込み向け高信頼プラットフォームの開発
## ―軽量仮想マシンモニタとセキュリティフレームワーク―

### 1. 背景

近年、携帯電話やデジタル家電のような高機能組み込みシステムが高性能化・高機能化する一方、バグやセキュリティ問題を残したまま市場に出荷される例が増えつつある。これは組み込みシステムの進歩に伴い、その開発が著しく大規模化・複雑化しているためである。このような状況を受け、近年では組み込みシステムのプラットフォーム化が進められている。これは、ソフトウェアを再利用することによって、開発サイクルの短縮やソフトウェア資源の再利用を進め、開発コストを削減し、システム安定性を向上するためである。プラットフォーム化の一手法として、これまでデスクトップシステムやサーバシステムで用いられてきた汎用 OS を、組み込みシステムで利用する例が増加している。これは汎用 OS のソフトウェア資産の転用を目的としている。しかし、汎用 OS は組み込み機器での利用を前提に設計されていないため、多くは十分なリアルタイム性能を満たしていない。この問題を解決するために汎用 OS をリアルタイム処理に対応するよう拡張する取り組みがなされているが、結果としてシステムの複雑化、不安定化を招いている。また、現状のデスクトップシステムやサーバシステムが直面しているウィルスやワーム等のセキュリティ問題は、組み込みシステムにおいても同様に深刻な問題となりつつある。今後、組み込みシステムの開発規模は増大しつづけ、十分な信頼性を維持することはますます困難となると予想される。

### 2. 目的

本プロジェクトの目的は、次世代の高機能組み込みシステムを対象とした、高信頼プラットフォームの開発である。プラットフォームは、汎用 OS の高機能性を提供しつつ、リアルタイム性を保証することを目標とする。また、ソフトウェアの再利用性を高め、開発コストの低減と安定性の向上を目指す。さらに、外部ネットワークからダウンロードされるアプリケーションから汎用 OS を保護しつつ、アプリケーションの制約を柔軟に変更するセキュリティフレームワークを提供する。

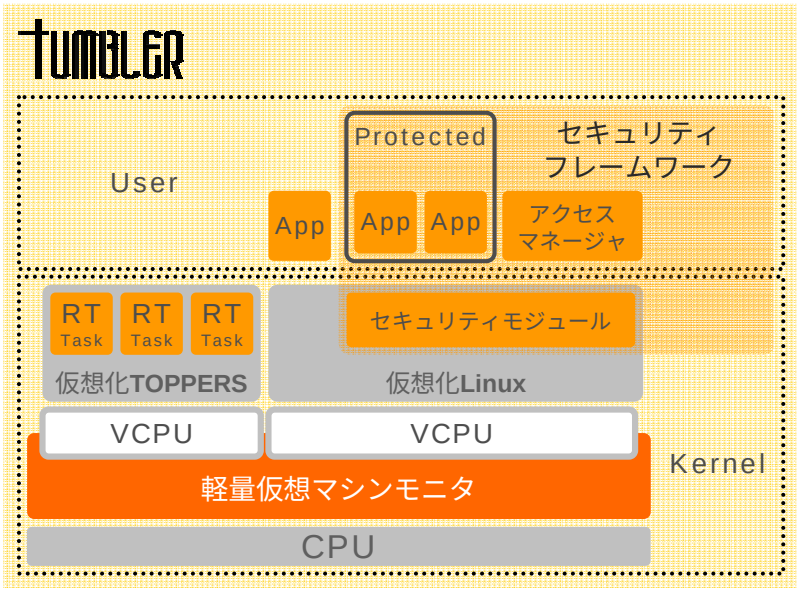


図 1: TUMBLER システム概観

## 3. 開発の内容

本プロジェクトでは組み込みシステム向け高信頼プラットフォーム TUMBLER を開発した。TUMBLER は主に軽量仮想マシンモニタ、セキュリティフレームワーク、リアルタイム OS（仮想化 TOPPERS）、汎用 OS（仮想化 Linux）によって構成される（図 1）。

### (1) 軽量仮想マシンモニタ

軽量仮想マシンモニタは単一のハードウェアを仮想的に多重化し、複数の OS を並列実行するソフトウェアである。ひとつのシステムでリアルタイム OS と汎用 OS を動作させることで、それぞれの OS の利点であるリアルタイム性と高機能性を両立させる。

表 1: 軽量仮想マシンモニタ提供機能

| | |
|---|---|
| OS 並列実行 | 複数の OS を単一のハードウェアで並列に実行する |
| 割り込み仮想化 | ハードウェア割り込みを、その要因に応じて指定した OS に分配する |
| OS 優先実行 | 複数の OS に優先度をつけて実行する。リアルタイム性が求められる OS を常に優先的に実行することで、そのリアルタイム性能を維持 |

表 2: 動作環境

| | |
|---|---|
| 対応ハードウェア | SH-2007, ASD-420 |
| 対応 CPU | SH-4A (SH7780, SH7770) |
| 対応 OS | TOPPERS, Linux |

### (2) セキュリティフレームワーク

セキュリティフレームワークは Linux に動的なリソース制約機能を提供する。図 1 に示されるように、セキュリティフレームワークはセキュリティモジュールとアクセスマネージャの連携によって動作する。

外部ネットワークからダウンロードされたアプリケーションはセキュリティモジュールによってリソースを制限される。しかし、外部ネットワークよりダウンロードされたアプリケーションの利用するリソースは、アプリケーションの種類や状況によって異なるため、リソース制約を静的に定義することはできない。そこで、アクセスマネージャがアプリケーションに割り当てるリソースの範囲を動的に変更する。

表 3: リソース制約対象

| | |
|---|---|
| ファイルシステム | ファイル、ディレクトリの単位でプロセスのリソースを制約 |
| ネットワーク | IP、ポート番号の単位でプロセスのリソースを制約 |
| シグナル | シグナル番号、送信元、送信先の単位でプロセスのリソースを制約 |

表 4: セキュリティフレームワーク提供機能

| | |
|---|---|
| リソース制約定義 | ・各リソースの制約条件を定義ファイルに記述する<br>・制約条件には正規表現を用いることも可能 |
| 動的な権限付与と剥奪 | ・アプリケーションの要求に応じて上記リソースへのアクセス権を動的に付与、剥奪する<br>・リソースへのアクセス禁止、許可は、ユーザの判断やソフトウェアの判断など、異なるアルゴリズムを用いることが可能 |

## 4. 従来の技術（または機能）との相違

・軽量仮想マシンモニタ

既存の仮想化技術はサーバシステムやデスクトップシステムでは広く用いられているが、組み込みシステムでの利用を前提としていない。そのため、リアルタイム性の点やハードウェアが仮想化をサポートしていない点で組み込みシステムでの利用に適していない。一方で本プロジェクトの軽量仮想マシンモニタは組み込みシステムをターゲットとして、OS のリアルタイム性能を維持するよう軽量に実装している。

また、これまでの組み込みシステムにおいても高機能性とリアルタイム性の両立を図るために、リアルタイム OS と汎用 OS を組み合わせて用いる例は存在するが、リアルタイム OS と汎用 OS の種類が固定されている。これに対して本プロジェクトの仮想マシンモニタはリアルタイム OS や汎用 OS についても選択可能となっている。

・セキュリティフレームワーク

既存の Linux のセキュリティ機構としては AppArmor や SELinux などの強制アクセス制御を行うセキュリティモジュールが存在するが、そのアクセス制御に関するポリシの設定は煩雑であり、一般ユーザが設定を独自に拡張するのは困難である。

一方、本プロジェクトでは、AppArmor による強制アクセス制御を基盤としたセキュリティモジュールと、その上で動的にアプリケーションのアクセス権限をコントロールするアクセスマネージャを用いることでより平易なアクセス制御機能を提供する。アクセスマネージャの動作の設定は一般ユーザに理解できる程度に単純化し、ユーザへの問いかけという形でアクセスマネージャの動作をカスタマイズできるようにしている。このような実装により、アプリケーションからアクセスさせてはならないリソースはアクセスモジュールの強制アクセス制御に基づいて保護し、そのほかのリソースに関しては、アクセスマネージャを通じ、アプリケーションの種類やユーザの意志によって動的にアクセス権限を付与することができるようになる。

## 5. 期待される効果

現在の高機能組み込みシステムは膨大な開発資源を投入してもなお、十分な信頼性を提供することが困難となりつつある。これはシステムの複雑化・大規模化が拡大しつづける一方で、システムを短いサイクルで開発する必要があるためである。高信頼プラットフォームは、OS やその上で動作するアプリケーションを再利用することで開発コストの低減を図る。また、異なる OS の利点をひとつのシステムに統合することにより、OS を機能拡張することなくシステムの要求事項を満たすことを可能とする。また、セキュリティモジュールにより従来の強制アクセス制御よりも柔軟性の高いセキュリティ機能を提供することで、安全なオープンプラットフォームの開発を支援することが期待される。

## 6. 普及（または活用）の見通し

リアルタイム制御と高機能アプリケーションが提供される高機能組み込み機器での利用を想定している。商用化に適したライセンスの下でオープンソースプロジェクトとして公開し、広く利用されることを期待している。

7．開発者名（所属）

杵渕　雄樹　（早稲田大学大学院　基幹理工学研究科　情報理工学専攻　博士後期課程）
湯村　悠　　（早稲田大学大学院　基幹理工学研究科　情報理工学専攻　修士課程）
香取　知浩　（早稲田大学大学院　基幹理工学研究科　情報理工学専攻　修士課程）
神田　渉　　（早稲田大学大学院　基幹理工学研究科　情報理工学専攻　修士課程）

（参考）　http://www.dcl.info.waseda.ac.jp/tumbler/