「実用化を目指した組込みシステム用
ディペンダブル・オペレーティングシステム」
平成 18 年度採択研究代表者

H22 年度
実績報告

前田 俊行

国立大学法人東京大学 大学院情報理工学系研究科・助教

# ディペンダブルシステムソフトウェア構築技術

## §1. 研究実施の概要

計算機（携帯電話、PDA、PC 等)、そして計算機を相互に接続するネットワークは、既に広く一般に普及し、もはや生活に欠かすことのできない社会基盤となった。このため、これらの計算機上で動作するソフトウェアのディペンダビリティを向上することが非常に重要な課題である、ということが既に広く認識されている。

しかしながら、計算機システムで最も基礎的で重要なソフトウェアであるシステムソフトウェア（オペレーティングシステムなど）は、依然として 30 年以上前に考案された C 言語や、アセンブリ言語等の安全でない言語を用いて構築されており、その安全性・信頼性には大きな疑いがあると言わざるを得ない。また現実に多くの問題、例えばシステムの異常停止、情報漏洩等の問題が発生している。

これに対し本研究では、近年目覚しい発展を遂げた静的プログラム解析技術（プログラムを数学的理論に基づいて解析することで、プログラムを実行することなく、その性質を知る技術)、特に型理論とモデル検査理論に基づき、システムソフトウェアの構築・検証技術を実現することで、システムソフトウェアの安全化・高信頼化を目指す。より具体的には、(1) システムソフトウェアを記述可能な型付きアセンブリ言語の設計・実装 (2) C 言語から型付きアセンブリ言語への変換器の設計・実装 (3) モデル検査技法に基づくシステムソフトウェアの解析、の 3 つについて研究を行う。

本年度は、昨年度までに作成した、システムソフトウェアを記述可能な型付きアセンブリ言語や C 言語から型付きアセンブリ言語への変換器の実装、また C 言語プログラムを対象としたモデル検査器の実装を用いて、他研究チームが作成したディペンダビリティ支援機構の検証を試みることで、現在の実装や理論の問題点や限界などの調査・検討・改善を進めた。また、上記の実装・理論とプログラム実行時の動的検査機構（ロギングやモニタリング機構など）との連携手法について検討を行った。更に、予期せぬシステム障害（Open Systems Failure）からの迅速復旧や問題修正等を行うことを目的とするシステム開発・運用プロセス全体の中において、上記の実装・理論がどのような位置づけにあるか、またどのような役割を果たすべきかについて議論・検討を行った。

次年度は、実際のシステムソフトウェア開発者による利用を念頭に、これら検査器の実装の利便性・完成度を更に上げることを目指す。また、システム運用・開発プロセス全体の中における検査器の位置付けを明確にする議論を更に進め、運用時のロギング機構やモニタリング機構、開発時のテスト機構、またより広い意味でのシステム分析手法（D-Case）との連携について理論検討・試験実装を行う。

## §2. 研究実施体制

（1）東京大学グループ

①研究分担グループ長：前田 俊行（国立大学法人東京大学大学院情報理工学系研究科・助教）

②研究項目

(1) システムソフトウェアを記述可能な型付きアセンブリ言語の設計・実装
(2) C 言語から型付きアセンブリ言語への変換器の設計・実装
(3) モデル検査技法に基づくシステムソフトウェアの解析

## §3. 研究実施内容

(1) システムソフトウェアを記述可能な型付きアセンブリ言語の設計・実装

本年度は、昨年度までに作成した型付きアセンブリ言語のプロトタイプ実装を用いて、我々の型付きアセンブリ言語の理論・実装の問題点について検討を行った。具体的には、他研究チームが作成したディペンダビリティ支援機構のソースコードやLinuxカーネルのデバイスドライバのソースコード等を我々の型付きアセンブリ言語へ変換することを試み、またプロトタイプ実装を用いて型検査を試みた。この結果、連結リスト構造(またはそれに似たデータ構造)をループ処理によって操作するようなプログラムの変換・型検査を我々の型付きアセンブリ言語で上手く扱えないことが分かった。この問題の解決に際して、分離論理の概念が有用であることが分かったため、この概念を導入した型システムの形式的定義・議論を行った。

また、予期せぬシステム障害(Open Systems Failure)からの復旧・問題修正等に対して、型検査技術がどのような役割を果たせるかについて、システム開発・運用プロセスの観点から議論・検討を行い、ホワイトペーパー等の形で発表した。具体的には、外部要因の変化や利用者等の要求の変化、事前に想定できなかった未知の障害等に伴って、システム分析(D-Case 等)の修正・改善が行われ、システムのソフトウェアの修正・新規作成が必要となった場合に、基礎的な安全性(プログラムが不正なメモリ操作が行わないことなど)を短時間で検証できるという型検査技術の特性を活かして、ソフトウェア開発者がソフトウェアの修正を行うのに追従して、基礎的な安全性を継続的に保証することができる。これにより、プログラム開発者は、些細なプログラミングミスなどに煩わされること無く、ソフトウェアの修正に集中することができる。一方、システム分析の修正に伴うソフトウェアの修正が期待通りに行われたかどうかの検証・テストは、モデル検査技術やベンチマーキング技術を用いて行うことができる((3)を参照)。

(2) C 言語から型付きアセンブリ言語への変換器の設計・実装

本年度は、昨年度までに作成した C 言語から型付きアセンブリ言語への変換器のプロトタイプ実装を用いて、我々の変換手法の理論・実装の問題点について検討を行った。具体的には、他研究チームの作成したディペンダビリティ支援機構のソースコードや Linux カーネルのデバイスドライバのソースコード等を我々の変換器で変換することを試みた。その結果、型付きセンブリ言語へ変換したプログラムとそれ以外の部分との相互の関数呼び出しやデータ受け渡しの処理を記述する手間が無視できないことが分かったため、この処理を行うプログラムを自動的に生成する手法について設計を行った。

また、動的検査技術、特にロギング機構(モニタリング機構)との連携について理論検討を行った。具体的には、プログラム中にログデータの取得を行う箇所を明示的に指定する場合において、冗長なログを取得しようとしている箇所を検出する手法や、変換器に実装した動的検査コード挿入の仕組みを応用して、ログデータを取得するコードをプログラム中に自動挿入する手法について検討を行った。

(3) モデル検査技法に基づくシステムソフトウェアの解析

本年度は、昨年度までに作成したモデル検査器の実装の改善を行った。具体的には、他研究チームの作成したディペンダビリティ支援機構のモデル検査を、実際に他の研究チームのメンバーに利用してもらうことで、モデル検査器の機能・性能・利便性に関する問題点の検討を行った。この結果、連結リストを用いるような、ポインタ周りの操作を頻繁に行うようなプログラムにおいて、検査の結果が期待通りにならなかったり、検査時間が長引いたりすることが分かった。この問題に対処するため、ポインタ解析を強化したモデル検査器の検討と、またそれにもとづく試験的な実装を行った。また、仕様記述言語において、同期ロックで保護されたデータに関する仕様の記述が上手く表現できないということが分かったため、この点を改良する仕様記述の方法についても理論検討・設計を行った。

さらに、検査に要する時間を短縮するための差分モデル検査(プログラムの修正に際して、全体を再検査するのではなく、部分的に再検査するような手法)の理論的背景として、自己適応計算の理論にもとづいて、通常どおり記述されたプログラムを自己適応計算的な形式のプログラムに自動的に変換する手法についての理論的検討・設計を行った。

また、予期せぬシステム障害(Open Systems Failure)からの復旧・問題修正等に対して、モデル検査がどのような役割を果たせるかについて、システム開発・運用プロセスの観点から議論・検討を行い、ホワイトペーパー等の形で発表した。具体的には、外部要因の変化や利用者等の要求の変化、事前に想定できなかった未知の障害等に伴って、システム分析(D-Case 等)の修正・改善が行われ、システムのソフトウェアの修正・新規作成が必要となった場合に、モデル検査器の利用者が指定した比較的詳細な仕様を、時間を掛けて検査するというモデル検査技術の特性を活かして、システム分析(D-Case)の修正から生じるソフトウェアに対する要求の修正・追加を、モデル検査器に与える仕様として可能なかぎり表現することで、プログラム開発者が行ったソフトウェアの修正が、システム分析の修正から期待される通りに行われたかどうかを検証することができる。またモデル検査器で検証できないような性質(性能要求など)やシステム全体として修正が正しく行われたかどうかの確認には、ベンチマーキングなどのテストで対応する必要がある。

## §4. 成果発表等

### (4－1) 原著論文発表

●論文詳細情報

1. Toshiyuki Maeda and Akinori Yonezawa, “Typed Assembly Language for Implementing OS Kernels in SMP/Multi-Core Environments with Interrupts”, In Proc. of the 5th International Workshop on Systems Software Verification, Oct. 6, 2010. (URI: http://www.usenix.org/events/ssv10/tech/full_papers/Maeda.pdf)

2. Junya Sawazaki, Toshiyuki Maeda, and Akinori Yonezawa, “Implementing a Hybrid Virtual Machine Monitor for Flexible and Efficient Security Mechanisms”, In Proc. of the 16th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2010), pp. 37-46, Dec. 14, 2010.
(DOI: http://doi.ieeecomputersociety.org/ 10.1109/PRDC.2010.32)
3. Toshiyuki Maeda, Haruki Sato, and Akinori Yonezawa, “Extended Alias Type System using Separating Implication”, In Proc. of the 6th ACM SIGPLAN Workshop on Types in Language Design and Implementation (TLDI 2011), pp. 29-42, Jan. 25, 2011. (DOI: http://dx.doi.org/10.1145/1929553.1929559)

**（4－2） 知財出願**

① 平成22年度特許出願件数（国内 0 件）

② CREST 研究期間累積件数（国内 0 件）