平成 21 年度
実績報告

「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

前田 俊行

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に関する研究

§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、前年度より引き続き、本研究領域全体で設計・実装中のディペンダビリティ支援のためのOS機構（P-Bus・P-Component）の検査・検証を目標として、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プロトタイプ実装や、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、また C 言語プログラムを対象としたモデル検査器のプロトタイプ実装を進めた。またこれらのプロトタイプ実装を用いて他研究チームが作成した幾つかの P-Component に対して検証を試み、その結果実際にコード中の重要な問題点を幾つか発見することができた。次年度は、これらのプロトタイプ実装の完成度を更に上げ、更に多くの P-Component の検査を行うことを目指

す。またプログラムの静的検査・動的検査と、実行時のディペンダビリティ支援機構の連携手法について更に検討を進める。

§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(東京大学、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

(文中に番号がある場合は(4－1)に対応する)

(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、昨年度より引き続き、より現実的な型付きアセンブリ言語のプロトタイプ実装を行った。またこの実装を用いて幾つかのディペンダビリティ支援機構の検証を試みた。具体的には、他研究チームによって実装された幾つかの P-Component のソースコードを型付きアセンブリ言語へ変換したものに対して型検査を行い、メモリ安全性や制御フロー安全性の検証を行った。その結果、ある P-Component のメモリ初期化部分に問題があることを検出することができた。この問題は、一見正しく動作しているように見えて、メモリの使用状況によっては稀に致命的な障害を生じる可能性があるという比較的深刻なものであったため、プロトタイプ実装の有効性を部分的に示せた上、ディペンダビリティ支援機構(P-Component)そのもののディペンダビリティの向上に貢献できた。また型付きアセンブリ言語の静的型検査に要する実行時間は数秒～数十秒の範囲であった。

また、この結果を他の検査ツールと比較するために、DEOSC (ディペンダブル組込み OS 研究開発センター)で実際に商用検査ツールを購入し比較検討したところ、上述の我々が検出した問題を検出することはできなかった。

(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、昨年度より引き続き、C 言語から型付きアセンブリ言語への変換器のより現実的なプロトタイプ実装を行った。またこの実装を用いて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変換を試みた。具体的には、他研究チームによって実装される幾つかの P-Components のソースコードの一部を、この変換器のプロトタイプ実装を用いて型付きアセンブリ言語の型検査器のプロトタイプ実装で検証できる形式に変換した。この変換では、メモリ安全性等を保証するために、実行時に動的検査を行うためのコードが挿入されるが、これに伴うオーバーヘッドは、実行時間に

関して数倍～十倍程度であった。また、この動的検査コードとディペンダビリティ支援機構との連携について、基礎的な検討を行った。

(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、C 言語で記述されたプログラムの安全性を検証するためのモデル検査器のプロトタイプ実装を行った。またこの実装を用いて、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検証を試みた。具体的には、他研究チームによって実装された幾つかの P-Components のソースコードに対して、ロックの整合性や P-Bus API の使用方法の正当性等についてモデル検査を行った。この検査を行うために、他研究チームと連携して、P-Bus API 仕様を記述するための仕様記述言語を策定し、また実際に仕様記述言語を用いて P-Bus API 仕様を記述した。この P-Bus API 仕様の記述に際しては、P-Bus や Linux カーネルの実装に関する一定の知識が必要であったため、他研究チームの協力の下、作業を外部委託することで効率的に実施した。この検査の結果、あるP-Component のロックの獲得・解放処理やタイマーの初期化・終了処理に問題があることを検出することができた。この問題はシステムの動作が停止したり、メモリ中のデータを破壊したりする深刻なものであり、更に稀なエラー処理時にのみ生じる、通常のテストでは検出することが難しい問題であったため、プロトタイプ実装の有効性を示せた上、ディペンダビリティ支援機構そのもののディペンダビリティの向上に貢献できた。なお、この検査に要する実行時間は、約三千行のコードに対して数分～三十分程度であった。

また、この結果を他の検査ツールと比較するために、(1)と同様に、実際の商用検査ツールと比較検討したところ、上述の我々が検出した問題を検出することはできなかった。ただし、当該商用検査ツールの検査実行時間は概ね数分の範囲にとどまっており、利便性を考慮してあえて検査の精度を落としている可能性もある。また、ツールの使い勝手に関しては、商用検査ツールに一日の長があるが、使い勝手の向上によって他研究チームからのフィードバックを得易くするという研究開発の観点から、比較的広く用いられているGUI統合開発環境との連携機能を外部委託によって実装した。

また、モデル検査をより現実的なシステム開発環境で行えるようにするために、プログラムの安全性検証を分割して複数計算機上で実行するための手法の実装・実験を行った。具体的には、プログラム中の特定のデータに注目して処理を分割することで、均等に処理を分割し、かつ、計算機間の通信頻度・通信量を低く抑える方法を検討した。更に、実際にクラスタ計算機システムを用いて実験を行い、その有効性・問題点の検討も行った。また、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差分モデル検査手法について基礎的な理論の検討を行った。

§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, “Writing an OS Kernel in a Strictly and Statically Typed Language”, Lecture Notes in Computer Science Vol. 5458, pp. 181-197, Springer, 2009. DOI: 10.1007/978-3-642-02002-5_10