研究課題別事後評価結果

1. 研究課題名： ディペンダブルシステムソフトウェア構築技術

2. 研究代表者名及び主たる研究参加者名（研究機関名・職名は研究参加期間終了時点）：
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）

3. 研究実施概要

本研究は、近年発展を遂げた静的プログラム解析技術のうち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・高信頼化を目指した。既存のソフトウェア高信頼化の理論研究では、実際的な議論や現実的な検証ツールの開発といった点が不十分であったが、本研究では既存の C 言語やアセンブリ言語等を対象に、一般の開発者にとって実際に実用的な検証ツールを開発することも目指した。

当初の研究構想には無かった、オープンシステムの考え方、すなわち、本質的に未知の障害要因が必ず存在し、事前に全ての障害要因に対処することは不可能であると仮定したシステムの捉え方において、プログラム解析技術を有効に適用する手法の考案も行った。

上記を達成するため、本研究では以下の三つの研究を行った。

(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

型付きアセンブリ言語とは、アセンブリ言語のレベルで型理論を応用することにより、メモリの安全性等を保証できる言語であるが、従来の型付きアセンブリ言語は複数のプログラムが同時に非同期に実行される環境を考慮しておらず、実用的なシステムソフトウェアに応用することは困難であった。

そこで本研究では、複数のプログラムが同時に非同期に実行されるような環境でも、メモリ安全性や制御フロー安全性を保証できるような型付きアセンブリ言語の設計・実装を行った。これを用いてシステムソフトウェアのメモリ管理機構やプロセス管理機構等を構築することで、システムソフトウェアの安全性・信頼性を向上できる。

(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

型付きアセンブリ言語などの安全なプログラミング言語を用いてソフトウェアを構築すれば、システムは安全になるが、既存のソフトウェアをその言語で書き直さなければならない。しかし、全てのソフトウェアを新たに異なる言語で書き直すのは現実的では無い。

そこで本研究では、既存の C 言語のソースコードをそのまま（もしくは必要であれば C 言語に僅かな拡張・注釈を加えることによって）ほぼ自動的かつ小さなコストで、型付きアセンブリ言語へ変換する手法の設計・実装を行った。これにより、既存の C 言語プログラムに対して型付きアセンブリ言語の型検査を応用することが可能になる。

(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

型付きアセンブリ言語などの安全なプログラミング言語を用いることで、ソフトウェアのメモリ安全性と制御フロー安全性を保証することができるが、現実的なシステムソフトウェアを考えると、より高度で複雑な安全性を保証することが必要な場合がある。

そこで本研究では、C 言語を用いて構築されたシステムソフトウェアに対しても現実的に適用可能なモデル検査手法・ツールの研究開発を行った。具体的には、システムソフトウェア、またそのシステムソフトウェアを利用するプログラムが満たすべき性質を、モデル検査器が直接解釈できる形式で記述し、これを用いてシ

ステムのモデル検査を行うことを目指した。

これら三つの研究成果によって、実際に他研究チームの作成した幾つかのプログラム等の検証が行える程度の実用性を持つツールを実現することができた。また他の商用プログラム検査ツールでは検出できなかったようなバグを検出することもできた。

4. 事後評価結果

4－1. 研究の達成状況及び得られた研究成果（論文・口頭発表等の外部発表、特許の取得状況等を含む）

本研究は、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・高信頼化を目指した。既存のソフトウェア検証ツールでは、並列プログラムが取り扱えない事や、C 言語やアセンブリ言語で書かれたプログラムに適応できないなど、実用性の面で多くの問題を抱えていた。本研究ではこれらの問題を解決するため、型理論とモデル検査理論にもとづいてプログラム検証システムを構築した。

具体的には、(1)複数のプログラムが同時に非同期に実行されるような環境でも、メモリ安全性や制御フロー安全性を保証できるような型付きアセンブリ言語の設計・実装を行った。(2)既存の C 言語のソースコードをそのまま（もしくは必要であれば C 言語に僅かな拡張・注釈を加えることによって）ほぼ自動的かつ小さなコストで、型付きアセンブリ言語へ変換する手法の設計・実装を行った。(3)C 言語を用いて構築されたシステムソフトウェアに対しても現実的に適用可能なモデル検査手法・ツールの研究開発を行った。

これらの研究はもともと個別研究テーマとして提案されていたものであるが、プログラム開発における利用と、そのエビデンスの D-Case への記録による説明責任の向上の手段として DEOS プロセス並びにアーキテクチャの考え方に整合した形でのツール作りとなり、本領域に貢献した。

これら三つの研究成果は多くの原著論文並びに国際会議論文として発表された。関連する特許申請の準備が行われている。実際に開発したツールを用いて、他研究チームの作成した幾つかのプログラム等の検証を行い、それらの実用性を示すことができた。この研究分野ではすでにいくつかの商用プログラム検査ツールが存在するが、それらでは検出できなかったようなバグを検出することもでき、本研究の成果を示すことができた。

一方で、「実用性」のレベルについては、技術の実用性を超えたツールとしての実用性が期待されていたが、これについては今後の課題として残された。

4－2. 研究成果の科学技術や社会へのインパクト、戦略目標への貢献

ソフトウェア検証技術を並列プログラムや C 言語やアセンブリ言語で書かれたシステムプログラムに応用可能にした点は高く評価できる。また、ソフトウェア検証技術を本研究領域の方針に合わせた形でツールとして実現し、いくつかのシステムプログラムに実際に適用してプログラムの不具合を発見したことは評価できる。ツールとしての実用化にはまだ時間がかかり、大きな社会的なインパクトを与えるに至っていない。

4－3. 総合的評価

ソフトウェア検証技術を並列プログラムや C 言語やアセンブリ言語で書かれたシステムプログラムに応用可能にした点は高く評価できる。研究成果は多くの原著論文並びに国際会議論文として発表されている。実際に開発したツールを用いて、他研究チームの作成した幾つかのプログラム等の検証を行い、それらの実用性を示すことができた点や、既存の商用プログラム検査ツールでは検出できなかったようなバグを検出することができたことは本研究の成果である。一方で、「実用性」のレベルについては、技術の実用性を超えたツールとしての実用性が期待されていたが、これについては今後の課題として残された。以上、研究領域の趣旨に照らし、妥当な成果が得られていると考えられる。