論文の内容の要旨

論文題目

Writing an Operating System with a Strictly Typed Assembly Language
(厳密に型付けされたアセンブリ言語を用いたオペレーティングシステムの記述)

氏名　　前田　俊行

近年の静的プログラム解析技術、特に型理論は、ソフトウェア開発のための基礎的な技術として既に広く用いられている。例えば、多くの実用アプリケーションが厳密に型付けされたプログラミング言語(Java、 C#、Objective Caml など)で記述されている。これら厳密に型付けされた言語の長所の一つは、それらの言語で記述されたプログラムの実行は失敗しない、すなわちプログラムが安全であるということが、型検査によって保証できる点である。

ところが、これら型理論の成果が未だに有効に応用されていないソフトウェアが存在する。それはオペレーティングシステム(OS)である。従来の OS は C 言語やアセンブリ言語などの厳密に型付けされていない言語、もしくは全く型付けされていない言語で記述されてきた。このため、OS の安全性を保証・検証することは著しく困難であった。

にもかかわらず、OS が厳密に型付けされた言語で記述されてこなかった理由の一つは、不可能だと信じられてきたことにある。これは、厳密に型付けされた言語では、OS の重要な機能であるメモリ管理機構やマルチスレッド管理機構を記述することができないように思われるためである。

これに対し本論文では、厳密に型付けされた言語を用いて OS を記述することが可能なことを示す。具体的には、OS の重要な機能(メモリ管理機構やマルチスレッド管理機構)が記述できる程度に柔軟でかつ表現力のある、新しい型付き言語を示す。この言語の鍵は、可変長配列(実行時まで長さが分からない配列)、明示的なエイリアス追跡、変数間の整数制約を型システムで直接サポートしている点にある。このため、現実的なメモリ管理機構(例: malloc/free)やマルチスレッド管理機構が記述可能となっている。また、本論文ではこの言語を用いて新たに記述した OS のプロトタイプ実装についても述べる。

この言語の型検査器で保証される安全性はメモリ安全性(プログラムが不正なメモリアクセスをしないこと)と制御フロー安全性(プログラムが不正にコードを実行しないこと)である。より複雑で洗練された安全性は、本論文で示す型システムを拡張することで保証可能となると考えられるが、これは本論文では扱わない。