POP Seminar

Why Operating-System Developers Write Barbaric Code and How to Verify It None the Less

Marcus Völp
Technische Universitat Dresden

Abstract:

Despite constant progress in languages, operating-system developers are typically rather prudent in adapting new features and languages and stick with languages such as C / C++ that are known to be unsafe from a type-safety perspective. Even worse, operating systems often contain code that explicitly violates type correctness and that often is not even conforming to the C / C++ standard. Drawing several examples from L4-family microkernel, I will try to defend some of the decisions that resulted in such barbaric code and present some ideas for verifying it.

I will briefly introduce our concept of well-behaving memory, which is at the root of our C++ semantics for verifying kernel code. I will talk about type sensitivity, which is our slightly different view on type and memory safety and highlight directions where our kernel will be moving and how we may verify it when it arrives there.

Biography:

Marcus Völp is a post doc at the Technische Universität Dresden, Germany and research group leader of the ESF young researcher group IMData. His research focuses on microkernels and microkernel-based systems with an emphasis on combining system aspects such as real time and information-flow security and on formally verifying microkernel based systems.

http://os.inf.tu-dresden.de/~voelp/

Friday, April 5, 2013
Gates Hillman Center 6501
2:00 PM – 4:00 PM