Felix S. Klock II

42 Pinecliff Dr, Marblehead, MA, 01945 USA, phone: +1 857.472.3757, e-mail: felix.klock@gmail.com

Objective

"Stop the bugs; save the world." Innovative software development, leveraging my experience in compiler and language runtime technology, in a challenging environment with an enthusiastic, smart, and respected peer group.

Software Development Skills & Interests

Programming languages: runtime design, memory management, JIT, static analysis
Software engineering: functional programming, debugging tool design, CS education
Languages: Rust, C/C++, Scheme/Lisp, Java, Python, C#, assembly (x86, ARM).

Education

Northeastern University, CCIS, Boston, MA **2003 – 2010**
Doctor of Philosophy in Computer Science

Massachusetts Institute of Technology, Cambridge, MA **1996 – 2001**
Bachelor of Science in Computer Science, 2000
Master of Engineering in Electrical Engineering and Computer Science, 2001

Professional Experience

Mozilla, Paris, France and remote **2012 – August 2020**
Staff Research Engineer, Rust language

Roles: (1) Rust compiler team co-lead, (2) Rust compiler developer, and (3) Language design team member.

Project home page: `https://www.rust-lang.org/`
Project history: `http://github.com/rust-lang/rust/`
Achievements: Compiler team management, triaging issues and chasing down bugs. Spun up working groups: *wg-prioritization* (delegating compiler bug triage) and *wg-incr-comp* (improving incremental compilation in Rust). Oversaw non-lexical lifetimes (NLL; Rust RFC 2094) development and staged migration from lexical to NLL. Designed future-incompatibility reporting (Rust RFC 2834), user-defined destructors with type and lifetime parameters (Rust RFC 769), non-zeroing dynamic drop (Rust RFC 320), and Rust's user-defined allocators (Rust RFC 1398).

Adobe Systems Incorporated, Waltham, MA **2010 – 2012**
Computer Scientist, Actionscript 3 Virtual Machine (aka Tamarin) for Flash Runtime

Roles: (1) Memory Management and Garbage Collection (GC) expert, (2) JSON, Array, ByteArray libraries, (3) integration lead, and (4) cross-platform build.

Project history through May 2012: `http://hg.mozilla.org/tamarin-redux/`
Achievements: Improved telescoping GC inverse load-factor, reducing overhead from 20× to 4× mark/cons ratio. Implemented Native JSON integrated with AS3, and extended with serialization of public members of AS3 classes. Added JIT-support for efficient indexing of "simple dense" `ArrayObject`, yielding 10–20% speedup.

Northeastern University, Boston, MA **2005 – 2010**
Developer and maintainer of Larceny Scheme compiler and runtime system

Project home page: `http://www.larcenists.org/`
Project history: `http://github.com/larcenists/larceny/`
Achievements: Designed regional GC, with formal bounds on mutator utilizaiton (MMU). Revised Intel x86 backend to emit machine code in-heap. Developed and evaluated four alternative x86 calling conventions, yielding $\geq$ 10% speed boost. Implemented dynamic in-heap .NET bytecode emission for Common Larceny.

Northeastern University, Boston, MA **2003 – 2009**
Instructor of Record/Teaching Assistant

Green Hills Software, Santa Barbara, CA **2001 – 2003**
Software Engineer for End-User Compiler Product Development

Achievements: Implemented move-coalescing register allocation, data-load optimizations, and reassignment of zero-initialized and uninitialized arrays to blank static segment (bss).

Massachusetts Institute of Technology, Cambridge, MA **2000 – 2001**
Teaching Assistant

Assistant for 6.170, *Laboratory in Software Engineering*, Spring, Fall 2000, Spring 2001. Head Teaching Assistant Fall 2000.

MIT Laboratory for Computer Science, Cambridge, MA **1999 – 2001**
Undergraduate Research Assistant for Computer Architecture Group

Assisted with implementation of *FLEX* compiler for transforming Java byte-code to machine code .

MIT Media Laboratory, Cambridge, MA **1997 – 1998**
Undergraduate Research Assistant for Software Agents Group

Helped develop *Footprints*, a tool for visualizing navigation of users on Web.

Research Publications

Felix S Klock II, “A Declarative DSL for Customized Rendering of Text-Based Art”, in *Proceedings of the 2017 International Symposium on Practical Aspects of Declarative Languages*, PADL 2017, Paris, France January 2017.

Felix S Klock II and William D Clinger. “Bounded-latency regional garbage collection”, In *Proceedings of the 2011 Dynamic Languages Symposium*, DLS 2011, Portland, OR, 24 October 2011, pages 73-83.

William D Clinger and Felix S Klock II. “Scalable Garbage Collection with Guaranteed MMU”, In *Proceedings of the 2009 Workshop on Scheme and Functional Programming*, Northeastern University, Boston, MA, 22 August 2009

Felix S Klock II, “The Layers of Larceny’s Foreign Function Interface”, In *Proceedings of the 2009 Workshop on Scheme and Functional Programming*, Victoria, British Columbia, 20 September 2008

Selected Presentations

“Rust: Fearless at all levels” Algolia Search Party 2018
`https://www.youtube.com/watch?v=9QuI2Z0stbs&t=2755`
“Subtyping in Rust and Clarke’s Third Law” Rust Fest 2016
`https://www.youtube.com/watch?v=fI4RG_uq-WU`
“Rust: A type system you didn’t know you wanted” Curry On 2015
`https://www.youtube.com/watch?v=Q7lQCgnNWU0&t=152s`

Honors and Awards

Teaching Award, Northeastern University, 2008

Northern Telecom/BNR Digital Systems Laboratory Project Award, 2001

Interests

Cooking; reading, especially historical discussions of mathematics, logic, and language development; skiing; graphics programming