Felix S. Klock II

42 Pinecliff Dr, Marblehead, MA, 01945 USA, phone: +1.857.472.3757, e-mail: `felix.klock@gmail.com`

Objective

"Squash bugs; save the world": Innovative software development leveraging compiler and language runtime technology 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, Scheme/Lisp, Java, Python, C++, 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

Amazon Web Services, remote **October 2020 – December 2024**
Principal Engineer, Rust Platform Team

Rust Roles: (1) Compiler team co-lead, (2) Specification team co-lead (3) Language Design team member, (4) Compiler performance working group member.

Project home page: `https://www.rust-lang.org/`
Achievements: Async Rust development (`tokio-console` performance probe, re:Invent workshop tutorial). Developer education screencasts (`rustc` development tutorials, "debugging as a science," `rr` and `pernos.co`). Safety validation (AWS-internal distribution of `miri`, the Krabcake `valgrind` tool, Rust Contracts, crowd-sourced stdlib verification).

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

Achievements: Spun up working groups: *wg-prioritization* (bug triage), *wg-incr-comp* (incremental compilation), *wg-rustc-perf* (monitoring rustc performance). Oversaw non-lexical lifetimes aka "NLL" (Rust RFC 2094), and migration from lexical to NLL. Future-incompatibility reporting (Rust RFC 2834). User-defined destructors with generic types and lifetimes (Rust RFC 769). Non-zeroing dynamic drop (Rust RFC 320). 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: Garbage Collection (GC) expert, integration lead, 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 history: `http://github.com/larcenists/larceny/`
Achievements: Regional GC, bounding mutator utilization. In-heap dynamic machine code emission for x86 backend. Dynamic .NET bytecode emission for Common Larceny.

Northeastern University, Boston, MA **2003 – 2009**
Taught Programming Languages, Intro to CS, and Program Design Paradigms

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

Achievements: Move-coalescing register allocation; data-load optimizations; reassignment of zero- and uninitialized arrays to blank static segment (bss).

Massachusetts Institute of Technology, Cambridge, MA **2000 – 2001**
Teaching Assistant, 6.170 "Laboratory in Software Engineering" for three semesters; Head Teaching Assistant Fall 2000

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

Assisted with *FLEX* compiler from Java bytecode to native 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

"Contracts for Rust, Revisited" Rust Formal Methods Interest Group 2024
`https://www.youtube.com/watch?v=j60gcdpHNaM`

"Fireside chat: Felix Klock and Greg Morrisett" ICFP 2023
`https://www.youtube.com/watch?v=msw47HLb3zo`

"Krabcake: A Rust UB detector" Rust Verification Workshop 2023
`http://pnkfx.org/presentations/krabcake-rust-verification-2023-april.pdf`

"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`

Honors and Awards

ACM Programming Languages Software Award, 2024, for contributions to Rust

Teaching Award, Northeastern University, 2008

Interests

Cooking; skiing; historical development of logic and natural language.