Felix S. Klock II

608 Tremont Street Unit 2, Boston, MA 02118, cell: 857.472.3757, e-mail: felix.klock@gmail.com

Objective

To apply compiler and runtime system implementation experience to innovative software development, in a challenging environment with an enthusiastic, smart, and respected peer group.

Software Development Skills & Interests

Programming language implementation: virtual machine/runtime design, memory management, just-in-time (JIT) compilation, static analysis

Software engineering: functional/declarative programming, system modeling tools, debugging tool design, computer science education

Languages: Scheme/Lisp, C/C++, Java, C#, Intel x86 assembly, FORTH, ARM/Thumb assembly.

Education

Northeastern University, Boston, MA **2003 – present**

Doctor of Philosophy, candidate

- Thesis: "Scalable Garbage Collection" (advisor: Prof. William D Clinger), expected Dec. 2009

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

- Thesis: "Architecture Independent Register Allocation" (advisor: Prof. Martin Rinard)

Professional Experience

Northeastern University, Boston MA **2005 – present**

Primary developer and maintainer of Larceny Scheme compiler and runtime system

Latest project development history viewable at `http://larceny.ccs.neu.edu/trac/`

- Designed and implemented regional garbage collector, eliminating pauses of unbounded length and introducing proven guarantees for minimum mutator utilization (MMU). This is novel and important: related research products either make *no* MMU guarantees at all, or *only* provide MMU guarantees when coupled with a behavioral model.
- Revised native Larceny Intel x86 backend to emit machine code directly to runtime heap dynamically, making compiler invocation transparent to client programmer, easing experimentation with code generation strategies, and improving system usability.
- Extended Larceny's Scheme/C Foreign Function Interface with procedure marshaling and C header processing, enabling linkage to native host libraries.
- Developed and evaluated four alternative conventions for native Larceny Intel x86 procedure invocation protocol, yielding $\geq 10\%$ speed boost over our prior calling convention.
- Implemented dynamic .NET bytecode emission for Larceny. Designed Scheme-based GUI toolkit and developed prototype source code editor and program debugger.
- Added automated build-test-benchmark infrastructure, replacing a tedious and fragile manual procedure with regular nightly process with web-based presentation of results.

Northeastern University, Boston MA **2003 – present**

Instructor of Record/Teaching Assistant

Instructor, CSG 111 *Principles of Programming Languages* (Spring 2008); Instructor, CSU 211 *Fundamentals of Computer Science I* (Spring 2006); Assistant, CSU 211 (Fall 2003, 2005; Spring 2009) and CSG 107/CS 5010 *Program Design Paradigms* (Fall 2008, 2009)

Green Hills Software, Santa Barbara, CA **2001 – 2003**

Software Engineer for End-User Compiler Product Development

- Added architecture-independent graph-coloring with move instruction coalescing as a prepass to legacy register allocator, improving object code size and speed, while avoiding development effort and risk of replacing the legacy codebase. Move coalescing also decreased overhead in subsequent compiler passes, yielding net reduction in average overall compilation time.
- Implemented data-load optimization, improving compiler output's performance to match performance of competitor's product.
- Implemented analysis reassigning zero-initialized and uninitialized arrays to blank static segment of object code, dramatically reducing object code size on client-provided benchmarks.
- Added peephole optimizations for ARM/Thumb backend, improving code size and speed.

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

Assistant for 6.170, *Laboratory in Software Engineering* (Spring and 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: designed data-flow analysis routines, StrongARM backend, and register allocation infrastructure.

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. Implemented hyperbolic graph display and path visualization tools. Designed communication protocol between application and central database. Advised on backend database design.

Research Publications

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, 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

Honors and Awards

Teaching Award, Northeastern University, 2008

Nothern Telecom/BNR Digital Systems Laboratory Project Award, 2001

Interests

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