Dr. Edd Barrett

An experienced systems programmer, computer scientist and open-source software developer.

www: https://eddbarrett.co.uk email: edd@theunixzoo.co.uk phone: (on request)
Address: (on request)

Employment

2023–today Research Fellow, King's College London

2013–2023 Research Associate, King's College London
Research interests include: JIT and AOT compilation, rigorous benchmarking methodology; programming language composition (building fast VMs which can run programs written in a mix of languages).

2009–2013 Teaching assistant, University of Kent
Part-time during PhD studies. C, UNIX, logic programming and Java. Undergraduate and masters.

2009–2013 Outreach project, University of Kent
Teaching programming fundamentals for schools (pupils aged 13-16) using Lego robotics.

2007–2008 Systems Administrator, Bournemouth University
Placement year as a part of undergraduate studies. UNIX systems administration (Solaris, Linux, OpenBSD) and in-house development.

2003–2005 Programmer, Tectonics Ltd.
Maintenance and reimplementation of a legacy stock management system, programming industrial CNC routers for kitchen fittings manufacturing, systems administration, IT support.

Education

2014 PhD in Computer Science from University of Kent
My thesis, Range Analysis of Binaries with Decision Procedures investigates the use of Boolean satisfiability (SAT) and linear programming for static reverse engineering of machine code.

2009 BSc Hons. Computing from Bournemouth University
First class with honours. My dissertation titled 3c: A JIT Compiler with LLVM describes the implementation of a JIT for a dynamically typed object-oriented programming language using LLVM.

2002 A- and AS-levels from Queen Mary's College, Basingstoke
Including Computing, Physics, Electronics and Mathematics.

Skills

Programming Languages I am an experienced programmer, currently most fluent in Rust, C, C++, Python and X86 assembler. I also have experience with Java, Prolog, Lua, Ruby, PHP, etc. In addition to using languages, I have also worked on compilers and VMs for languages: C/C++ (i.e. Clang/LLVM), Lua, Rust, Python, PHP and Prolog. I can learn new languages/paradigms as required.

Software Engineering The essential practices that accompany the code itself. Version control, testing, test reduction, bug reporting, debugging, code review, continuous integration, benchmarking, etc.

Computer Science Since 2009 I've worked in an academic environment, publishing (and reading) papers on both theoretical and practical topics including: language composition, code generation, benchmarking, tooling, abstract interpretation, optimisation problems, propositional logic, etc.

Systems Administration I've always enjoyed tinkering with UNIX/Linux. I self-host many services for my own personal use (e.g. mail and web servers). I also maintain benchmarking and continuous integration infrastructure for my team at King's College London.

Academic Service/Organisation

I've served on the committees of many computer science conferences, workshops and journals, including: ICOOOLPS, DLS, Dyla, COMLAN, JOT, SoCP, MoreVMs, ICW, <Programming>, MPLR, ISMM and VMIL. I've served on the organising committee of MoreVMs several times. I was publicity chair for ECOOP'19 and DLS'14.

I co-organised Barcamp Canterbury 2012–2014 and 2019.

Team Skills I'm a friendly and approachable person who enjoys working with others.

Software Contributions

The Yk JIT Ongoing work building a meta-tracing JIT similar to PyPy, but based on LLVM. Existing interpreters written in C can be lightly modified to add a multi-threaded JIT. The system comprises of a fork of LLVM (C++) and a JIT runtime written in Rust.

PyHyp An RPython VM that can JIT a mix of Python and PHP code. The system enables fine-grained language interoperability, allowing: cross-language scoping; passing objects between languages; cross-language exceptions, etc. This work was published at ECOOP'16.

Krun A benchmarking harness written in Python, designed to reduce the impact of confounding variables upon benchmark measurements. This tool was used in work that was published at OOPSLA'17.

OpenBSD Developer for an open-source UNIX distribution since October 2009. I work mostly on the ports tree, which involves patching and packaging third party software (TeX Live, Neovim, graphviz, Syncthing, ...). I've also dabbled in the kernel and a few base distribution userspace programs.

Dgen/SDL Implemented a debugger for an open-source games console emulator (just for fun).

Selected Publications

- E. Barrett, C. F. Bolz-Tereick, R. Killick, S. Mount and L. Tratt. Virtual Machine Warmup Blows Hot and Cold. *Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*, October 2017.
- E. Barrett, C. F. Bolz, L. Diekmann, L. Tratt. Fine-grained Language Composition: A Case Study. *European Conference on Object-Oriented Programming (ECOOP)*, July 2016.
- E. Barrett, C. F. Bolz, L. Tratt. Approaches to Interpreter Composition. *Computer Languages, Systems and Structures (COMLAN)*, December 2015.
- E. Barrett, C. F. Bolz, L. Tratt. Unipycation: A Case Study in Cross-language Tracing. *Virtual Machines and Intermediate Languages (VMIL)*, October 2013.
- E. Barrett. Range Analysis of Binaries with Decision Procedures. *PhD thesis*, March 2014.

Hobbies, Interests and Spare Time

I like to tinker with electronics and repair things to give them a new lease of life. I read science fiction novels, collect music and grow my own vegetables. I love camping with friends and family. I keep in shape by riding my bike and going to fitness classes at the gym. I speak Spanish and meet with a group weekly for conversation.

References

Professor Laurence Tratt
Department of Informatics
Strand Campus
King's College London
London, WC2B 4BG
laurence.tratt@kcl.ac.uk

Professor Andy King
School of Computing
University of Kent
Canterbury
Kent, CT2 7NF
a.m.king@kent.ac.uk

(Last updated November 6, 2025 – git version 88bd2a1)