

Vedant Kumar

vk@vedantk.com

Interests

Compilers and programming languages, operating systems, and computer graphics.

Experience

- Apple, 2015-, Compiler team (LLVM/Clang development)
- Apple, 2013-2014, Filesystems Team (Summer Intern, 2x)
 - Worked on the XNU kernel, CoreStorage, and HFS+.
 - Wrote randomized testers and `lldb` extensions to debug race conditions in XNU.
 - Top-five finalist in the Apple-wide software engineering intern competition (2013).
- UC Berkeley, 2015, Programming Languages and Compilers (Teaching Assistant, [class website](#))
- UC Berkeley, 2014, Advanced Operating Systems (Teaching Assistant, [class website](#))
 - Co-taught a discussion section, wrote labs, and wrote exam questions.
 - Wrote an EDF scheduler, a small filesystem, and a device driver for Linux (in a team).
- HP Fortify, 2012, Static Analysis Team (Summer Intern)
 - Designed, implemented, tested, and shipped an interprocedural constant propagator.
- Personal projects (C++, Python), 2009-2014. All source code is available [here](#).
 - Data structures: skiplist, heaps, trie, B+ tree, kd-tree, open-addressed hash table
 - *quotient-filter*: compact approximate membership filter, supports merging and deletions
 - *53otron*: primitive Lisp \rightarrow LLVM compiler, used to vectorize equations for a 3d visualizer
 - *auto-diagonalize*: optimization pass which converts linearizable loops into $\Theta(\log n)$ processes ¹
 - Graphics: a fast ray-tracer (*radiate*), a model controlled with inverse kinematics (*spike*)
 - Networks: an `epoll`-based server in C (see: [vedantk.com](#)), multi-threaded servers

Education

I graduated in May 2015 with a B.S in Electrical and Computer Engineering from UC Berkeley. I have taken the following relevant classes:

- Advanced Operating Systems
- Computer Architecture and Engineering
- Intro and Graduate CS Theory
- Integrated Circuits and Digital System Design
- Programming Languages and Compilers
- Discrete Math and Probability
- Graduate Programming Language Semantics
- Foundations of Computer Graphics

¹I demoed the optimization pass at the 2014 LLVM developer meeting: [slides](#), [longer writeup](#).