

ERIC ZHOU

(805) 832-7323 • ericfzhou@berkeley.edu • [zehric.github.io](https://github.com/zehric) • linkedin.com/in/zehric

C • C++ • x86 • Perl • Python • Java • Go

EDUCATION

August 2015 - May 2019 (Expected Graduation)

University of California, Berkeley

B.S. Electrical Engineering and Computer Sciences

GPA 3.93

Awards: Honors to Date • Dean's List • Eta Kappa Nu

Relevant Courses: **CS164** (Programming Languages and Compilers) • **CS162** (Operating Systems and Systems Programming) • **CS186** (Databases) • **CS170** (Efficient Algorithms and Intractable Problems) • **CS189** (Machine Learning) • **CS161** (Computer Security) • **EECS151** (Digital Design and Integrated Circuits) • **EE140** (Linear Integrated Circuits)

EXPERIENCE

Research

Robot Arm (Fall 2017 - Spring 2018)

I worked on firmware for an STM32 microcontroller that drive the motors for a completely custom robot arm, as well as the software that controls it. The boards run on ChibiOS, an open source RTOS. I worked on a team under the supervision of Professor Pieter Abbeel.

Internships

VLSI Intern at NVIDIA (Summer 2018)

I researched the increasing intensity of self-heating effects in the 7nm technology process and how it impacts electromigration, which is a main factor in the lifetime of a chip.

Software Development Engineer Intern at Amazon (Summer 2017)

I developed a web interface for Amazon Fresh internal usage that allows for safe and quick updates to merchant schedules. One of the impacts of this tool is that it increases the speed at which Fresh can launch in new regions. The application uses a Scala backend with an AngularJS frontend.

Software Intern at Rently (Summer 2016)

I created support for controlling Rently Keyless smart home devices (mainly locks) on the Amazon Echo. I created an Amazon Alexa Skill that forwards raw English text to a custom natural language parser.

Teaching

uGSI for CS162 (Operating Systems) (Fall 2018)

PROJECTS

Operating System (Ongoing)

My own operating system for x86, started from scratch, created mainly for fun and learning purposes. My current goal is to create a simple, monolithic kernel that features preemptive multitasking, along with a barebones text user interface, though this is subject to change.

Course Projects

Pintos (Spring 2018)

A small instructional operating system for x86. We implemented several schedulers, numerous essential system calls, and a filesystem based on the BSD FFS. Notably, I designed a buffer cache with a unique synchronization scheme that minimized unnecessary waiting without being overly complex.

CPU (Fall 2017)

A fully tested and functional 32-bit RISC-V CPU with a 3-stage pipeline and cache written in Verilog and pushed through the ASIC design flow with Synopsys Design and IC Compiler.