James Brisson
912 #B Romeria Drive; Austin, Tx 78757
(512) 497 9486 <theotherjimmy@gmail.com>

SKILLS

Programming Languages	Rust, C[++], Haskell, LaTeX Python, Bash, Make
Assemblies	ARMv7E-M (GAS), Intel i686 (GAS), Freescale 6812, TI TMS320C6000 DSP,
Software Development	Git, Merge Request workflow, GDB

EDUCATION

BS in Electrical Engineering	UT Austin in December 2013
Tech Areas	Computer Design, Communications/Digital Signal Processing
Notable Classes	Operating Systems Honors (using C), Real-time DSP Lab, Computer Architecture, Real-time Embedded Systems

Operating Systems:

- Modified the Linux kernel scheduler and implemented several kernel modules
- Developed an exokernel for i686 in C and asm; ext2 drivers, self-hosting, graphical

PROFESSIONAL EXPERIENCE

May 2019 - Nov 2024: Embedded Firmware Engineer at Arm:

- Benchmarked and improved TF-A memory footprint with custom tooling
- Snapshot debugger for Zepyhr with support for armv8m and TF-M
- Runtime debugger for Arm FVP including bridging to GDB and event logging
- Transitioned a Yocto Linux distribution from sysVinit to systemd
- Bench-marked and tweaked Kubernets for memory performance on an arm Linux single board computer

May 2016 - April 2019: Mbed OS Software Engineer at Arm:

- Extended build framework with new features while maintaining backwards compatibility
- Maintained an offline and online, multi-tenant IDE and testing infrastructure in Python
- Mentored open-source contributors to improve contribution quality and git history quality
- Developed testing infrastructure that ran about 30k tests for each pull request
- Automated parts of the Pull Request(PR) process and developed a contribution model to handle 100 PRs a week

March 2014 - May 2016: Research Assistant:

Working on several academic research projects, including buddythreads and bubbles, mentioned below.

Jan 2015 - May 2015 and Aug 2015 - Dec 2015: Teaching Assistant:

During Spring 2015 I was a TA for EE319K, Intro to Embedded Systems, and During the Fall of the same year I was a TA for EE379K, Operating Systems.

May 2013 - December 2013: Intern Silicon Labs:

- Automated build system creating patch-able 8051 ROM and automated patch making
- Created testing framework for pre and post silicon (simulation, FGPA emulation, evaluation)
- Wrote firmware RC oscillator calibration algorithm and several patches
- Developed waveform capture tool for firmware symbols on a simulated 8051 processor

PROJECTS

- Cmsis-pack-manager: A highly concurrent download utility for CMSIS Packs written in Rust and Python
- Moses: A bluetooth controlled holonomic robot and controller
- Automated framework for estimating side channel capacities of contention based channels
- RASLib: intro to robotics library targeted at the TI Stellaris/Tiva Launchpads
- Custom Keyboard, Dactyl, with custom layout and firmware in Mbed OS
- Intelligent ground vehicle software design

RESEARCH

Leg: An arm Emulator for fuzzing:

A prototype emulator foccussing on the bare minimum of hardware emulation to boot TF-A, the ability to snapshot and restore quickly and multi-threading. Leg emulates with both a simple interpreter and a JIT, with threads that fail to take the JIT lock falling back to the interpreter.

Buddythreads: Scheduler-Base Side Channel Defenses:

A modification to the Linux kernel that allows a process to request that it should always be scheduled simultaneously with another, "buddy" thread. This allows the buddy thread to make noise on shared resources that may be used for side channel attacks. I also developed and evaluated several methods for creating noise on these shared resources and showed that perfect information of the victim's leakage is sufficient to thwart attacks.

Submitted to ISCA 2016.