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

Skills

Programming Languages	Rust, C[++], Haskell, LaTeX Python, Common Lisp, Scheme, Clojure, Bash/Fish, elisp, Make
Assemblies	ARMv7E-M (GAS), Intel i686 (GAS), Freescale 6812, LC3, TI TMS320C6000 DSP,
Software Development	Git, Merge Request workflow, Vim, Make

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 assembly; ext2 drivers, self-hosting, graphical
- Implemented an RTOS for the ARM Cortex-M in C and assembly
- Compiled custom kernels, Linux and Android, with patching

Professional Experience

May 2019 - Ongoing: Embedded Linux Software Engineer at Arm:

- 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

Summer 2010: Outback Director BTSR:

- Managed 3 staffers leading a trek a week
- High adventure backpacking program
- Planned and tracked food and gear distribution across many campsites

Summer 2009: Scout Skills Director BTSR:

- Managed 3 staffers teaching classes
- Taught camping and outdoor skills
- Responsible for the teaching of 14 classes
- Lead toten chit and fireman chit sessions

PROJECTS

- Aura: An Arch Linux package manager wrapper written in Haskell
- 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 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
- Custom Keyboard, Mx Butterfly, with custom layout and firmware using TI Stellarisware
- Intelligent ground vehicle software design
- Remote controlled mobile couch with turn signals
- Robotics Booster-pack for TI Stellaris/Tiva Launchpads (PCB design)
- Planar image stitching algorithm using phase correlation
- QPSK transceiver

RESEARCH

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.

Bubbles Secure Sharing System:

A prototype security system that refactors the sharing out of mobile and web applications and makes it secure. Sharing is then done on a ‘Bubble’ level, where information is grouped by the user into a single package, or folder, that may be shared. Design principles of the system include:

- Principal of Least surprise: minimize changes to the programming environment of the developer
- Interface integration between the application running in the Bubbles system and the system itself
- Lightweight: The containers that provide the security for the Bubbles system should have minimal performance impact

submitted to Okaland 2016

COMMUNITY SERVICE

- Mentor for 2013 and 2014 UT RAS Robotathon, Region V, and Mercury teams
- Eagle Scout Project — build privacy fence for Humane Society of Williamson county
- Over 125 hours of community service through Boy Scout Troop 513

SOCIETY MEMBERSHIPS

- Eagle Scout
- IEEE Robotics and Automation Society UT student branch
- IEEE UT student branch

Recommendations available upon request