J. AUSTIN **SEIPP**

Austin, TX
aseipp@pobox.com

I am a software developer and infrastructure engineer with strong interests in low-level engineering, functional programming, software security, developer tools, hardware design, and the theory and practice of free software. Expert C and Haskell programmer; expert in package management and build systems, developer tooling, developer UX. Expert Linux admin/operations engineer, with 10+ years of contribution to NixOS Linux. Over 15 years of FOSS contribution experience across dozens of languages. Former production Haskell compiler engineer. Experience in RTL hardware design and development.

EXPERIENCE

OCT 2023 – JUNE 2024 (9 MO)

SENIOR HASKELL ENGINEER, KADENA.IO

I worked as a Haskell programmer on the core distributed blockchain system known as “Chainweb.” I spent most of my time doing code review, architectural work, and operations work. Deployed and maintained Gerrit as a highly effective code review tool for our team, improving developer velocity significantly. I also spent time educating the Eng dept. on better code review practices, better version control practices, and build integration.

AUGUST 2022 – DECEMBER 2022 (5 MO)

SENIOR CAD/EDA ENGINEER, GROQ INC

I worked as a software engineer with EDA engineers on the design and physical layout (“Physical Implementation”) of the Groq Tensor Streaming Processor. This largely revolved around automating third-party EDA tools (Cadence, Synopsys) through combinations of TCL, Python, Nix, and more, as my goal was to learn more about nanoscale semiconductor physics and design.

I also provided much advice to fellow engineers on developer operations (“DevOps”), infrastructure design, Nix tooling expertise, Haskell programming, and more.

2019 – 2020 (18 MO)

SENIOR ENGINEER, FASTLY INC

I worked on the compiler tools for Varnish CDN client configuration, gave internal talks on the theory of build systems and FOSS project contribution, and worked on the initial design of a distributed, eventually consistent data store for unannounced product enhancements.

2017 – 2018 (21 MO; CONSULTANT)

SENIOR ENGINEER, MYRTLE SOFTWARE

I worked on a Deep Learning compiler for an FPGA-based hardware accelerator using Haskell, designed circuits using Haskell, and worked on build systems (Nix, Haskell) and system software infrastructure for those accelerators (Linux device drivers, userspace libraries, *et cetera*). I also trained and helped junior engineers in both Nix and Haskell-related development practices.

2013 – 2017 (4 YRS; CONSULTANT)

HASKELL CONSULTANT, WELL-TYPED LLP

I developed and supported the open-source Glasgow Haskell Compiler. I was a lead developer, release manager, first point of contact, and involved in every aspect of the project: from design to bug triage to infrastructure. I overhauled the developer experience of the project, instituted the first widely accepted code review system, and did significant contributor onboarding. I helped apply sharper software engineering principles throughout the whole project. These initiatives resulted in a significant rise in talented, wonderful contributors to the project, a trend that continues today: including two who have inherited my prior title.

2012 – 2013 (16 MO)

SECURITY RESEARCHER, RAPID7 INC

I worked in a group dedicated to the discovery of remote vulnerabilities in hardware, software, and networking devices. I also reverse engineered actively exploited software flaws found in the wild, and did my own exploit development for popular software — including Firefox and Samba.

2009 – 2011 (2.2 YRS)

SOFTWARE ENGINEER, R1SOFT INC

Developer in the Backup Division of Idera Software. I collaborated with a small team to implement “transparent” copy-on-write backup software for existing Linux and Windows systems (ext3/xfs/NTFS) through a combination of kernel, server, and client programming.

OPEN SOURCE (NON-AUTHORITATIVE LIST)

2023 – CURRENT (1 YR)

CONTRIBUTOR, JUJUTSU VERSION CONTROL

I am an active developer on Jujutsu, a fast, friendly, and easy-to-use version control system (VCS) that is compatible with Git. While reusing the Git storage model, it has a completely rebuilt user interface and set of algorithms. While from prior art from the likes of Git, Mercurial, Google’s Piper, and Pijul, Jujutsu also offers many novel concepts such as the “working copy as a commit”, the “operation log”, and “first class conflicts” — and we have many enthusiastic and supportive users among a growing community.

My interest in Jujutsu currently revolves around UX, authoring new features, release integration procedures, developer support and experience, and community leadership.

2023 – CURRENT (+1 YR)

CONTRIBUTOR, BUCK2 BUILD SYSTEM

I am an active participant and contributor/community member for a new build system, Buck2, designed by Meta. Buck2 is a theoretically *and* practically sound, scalable, multi-language build system, inspired by research into incremental computation and build system design. My interest in Buck2 revolves primarily around “external” concerns to the Meta team, including release management, FOSS usability, distribution concerns, NixOS integration, developer UX, and much more.

2014 – CURRENT (8+ YRS)

CONTRIBUTOR, NIXOS LINUX & NIXPKGS

I am a long-time contributor and user of the NixOS Linux Distribution and Nixpkgs package set. Nixpkgs is one of the largest open-source software repositories available in the world, available for any Linux Distro. I currently maintain most EDA tools that are available; previously I maintained PostgreSQL, OpenJDK, worked on networking, time subsystems, security hardening, and dozens of other packages. I'm one of the top 50 committers of all time in the project.

2010 – 2017 (7 YRS)

CONTRIBUTOR, GLASGOW HASKELL COMPILER

I wrote code for the Glasgow Haskell Compiler as a contributor long before I was the lead maintainer; I helped author and integrate the initial work for compiler plugins, cross compilation, and other general "janitorial" duties like code cleanup, platform bugfixes, and patch review. I am still one of the all-time top contributors to the project; these experiences gave me insight into the "ground floor" of software engineering in FOSS projects.

2013 – 2017 (4 YRS)

ADMINISTRATOR, HASKELL.ORG

During my time as a contributor and lead of the Glasgow Haskell Compiler, I was also an active administrator of Haskell.org — I maintained servers, applied security patches, answered user queries, fixed broken websites, created and maintained new services, and retired old ones for a large and varied userbase.

SKILLS

- Strong interest in software security and low-level software engineering: CPU design, kernel engineering, etc
- RTL development: I am a user of "alt-HDLs" for Hardware Development. My current tool of choice is Bluespec.
- Strong interest in formal methodology tools, mathematics, and "computational system dynamics": model checking, theorem proving, state space design, invariant design through APIs, and much more.
- Active, working knowledge of cryptographic security principles and algorithm design. Implemented various IETF standards, model checked them, and written alternative specifications.
- Expert in functional programming, having been a Haskell programmer for over 15 years.
- Deep insight into the principles and theory of free software development, labor, and more, having been a free software developer my entire adult life.

OTHER INTERESTS

I'm actively interested in the design and development of friendly, high-scale developer tools and developer UX; notably build systems (Buck2), version control (Jujutsu), and code review (Gerrit, Phabricator). I spent years helping maintain, code review, contribute, and lead projects like the Glasgow Haskell Compiler, which gave me a deep appreciation for developer onboarding and UX.

I am interested in the design of distributed database systems, particularly cost-based query optimizers and compiler technology, and transaction management systems. I also have operational and development background in many relational databases (MySQL, PostgreSQL), but my interest in the theory is largely independent of that.

I deeply care about and spend time thinking about the labor relationships between free software, its users, developers, and the rise of “corporate free software” in the modern development age. This includes everything from the theory of free software itself and its historical underpinnings, to concrete strategies for engaging with users in a supportive and manageable way, to thinking about the working and extant class relationships between software authors and users.