Connor Taffe

(501) 606-1807 · cpaynetaffe@gmail.com · byteflame.org

EXPERIENCE

Apptegy January 2017
Software Engineer *Little Rock, Arkansas*

- Lead transition from monolithic Rails application to Ruby (Roda+Sequel) containerized microservices.
- Introduced memcached caching via Sequel ($\frac{1}{4}$ latency on average for cached requests).
- Introduced and implemented OpenAPI (Swagger 2.0) specifications during rearchitecture.
- Implemented CI (tests, style checking) and CD (docker build, push to registry) with Jenkins for rearchitecture.
- Developed and containerized internal services: HKP GPG key server (initially deployed for storing encrypted secrets in version control), OpenAPI documenation server, Jenkins build server, Matrix chat server, KaeruEra error tracker.

Acxiom September - December 2016
Entry Software Engineer *Conway, Arkansas*

- Big data processing with Apache Hadoop and Spark
- Explored optimization techniques using Apache Spark

All Electic Supply June - September 2016
Programmer *Little Rock, Arkansas*

- Heavy usage of PostgreSQL SQL, `psql`
- Use of Docker for testing based on production database backup
- Wrote helper programs in Go to send emails and perform regular queries
- Submitted accepted PRs to xTuple (ERP software)

Ensono (formerly Acxiom ITO) October 2015 - May 2016
Entry Applications Developer *Conway, Arkansas*

- Wrote and maintained software in C# for the .NET platform.
- Developed applications in Node.js, Golang, and Python for AWS Lambda.
- Versioned software in Git and maintained repositories in Bitbucket.
- Participated in Agile training and worked as part of a scrum team.

UALR EIT IT August - October 2015
Student Worker *Little Rock, Arkansas*

- Resolved technology related issues for staff and students.
- Tracked issues in ticketing system.
- Installed and Setup Microsoft Windows, including joining to Active Directory domain.

EIT SUPER, UALR May - August 2015
SUPER Scholar *Little Rock, Arkansas*

- Summer Undergraduate Program of Entrepreneurship and Research (SUPER).
- Creation of an Android application, written in Scala, to interface with a vehicle's OBD-II system via Bluetooth transmitter.

Emerging Analytics Center (EAC), UALR October 2014 - May 2015
Software Engineering Intern *Little Rock, Arkansas*

- Presented at IEEE VR 2015 on the integration of Unity 3D, Qualcomm's Vuforia, and Intel's OpenCV to create interactive Augmented Reality (Arkansas) applications (presentation).
- Used Unity 3D (scripting with C#) for 3D programming and model manipulation.

EDUCATION

University of Arkansas, Little Rock
Pursued a B.S. in Computer Science

- Vice President of the UALR instance of the Association of Computing Machinery; Fall 2016
- Substantail completed courses: Calculus I, II; Discrete Math; Linear Algebra. Data Structures and Algorithms, Computer Systems and Assembly Language, Operating Systems, Databases, Computer Organization, Monte Carlo Simulation, Independent study on Compiler Design, Artificial Intelligence, Langauge Structures, Computer Security.

Arkansas School for Mathematics, Sciences, and the Arts
Attended

- Substantial completed courses: Programming II, Physics I.

PERSONAL PROJECTS

birch August 2016

- IRC Bot in C
- Lexing/parsing of messages using state machine

Lispy September - November 2014

- Lisp-like interpreter written in Python
- Producer-consumer threading optimizes stages
- Lazy-evaluation of defined variables
- lambda functions and recursive lambdas
- EBNF formal definition of the language

utf8 / **utf8plus** February 2015

- utf8 parser written in C, and C++ wrapper.
- RFC 3629 compliant.
- Quick encoding/decoding of runes, rune validation, and string parsing.
- Reports errors within the rune and codepoint (`int32_t`) types using approprate non-valid values in C.
- Use of the `utf8::rune::exception` class to report errors in C++, or appropriate C++ standard exception.

bf November 2014

- Optimizing BrainFuck just-in-time compiler
- Emits `x86_64` instructions to executable `mmap`'d pages
- Producer-Consumer concurrent architecture with a REPL interface

TECHNICAL STRENGTHS

General Programming

Languages:	Ruby, Go, C, C++, Python, JavaScript, Typescript, (X)HTML, CSS, Scala, Java, C#, Rust, Php
Libraries:	Roda, Sequel, Rails, Vue.js, Android SDK, Bootstrap
Systems:	Hadoop, Spark
Tools:	Docker, Git, Make, Ninja, CMake, Gradle
Databases:	PostgresSQL
OS:	Linux, Windows