

JORDAN RICHARDS

Systems Software Engineer

📍 Orlando, FL

@ jordan@carbondoos.dev

☎ 4072849652

🌐 carbondoos.dev

📧 carbon_xiii

📄 jordan-alexander-richards



EXPERIENCE

Senior Software Engineer

RBC Capital Markets

📅 Apr 2018 – Sep 2023

📍 Orlando, FL

Senior software engineer working on the trusted execution components of an **always-encrypted offers service**.

- Developed a C++20 microservice from initial design to implementation and integration
- Helped integrate with Java microservices
- Cryptography: ECDH key exchange, key wrapping, encryption
- Microsoft SQL Server
 - Utilized the new feature, always-encrypted columns with secure enclaves
 - Worked with team at Microsoft to work out bugs in the ODBC driver
- Trusted Execution Environment: Built on top of **mystikos**
 - Tight memory and performance constraints
 - Built upon limited existing tooling for debug and analysis
 - Deep dive into Linux kernel internals to debug memory mapping in mystikos

Team Competition Programming

University of Central Florida

📅 Aug 2018 – Mar 2021

UCF varsity programming team member, participating in 3 ICPC regional contests.

- Challenging algorithmic design puzzles with tight constraints
- Developed and employed in-depth knowledge of a wide range of useful data structures.
- Experience implementing high-performance C++
- Developed skills working with a team under pressure

EDUCATION

B.S in Computer Science

University of Central Florida

📅 May 2022

📍 Orlando, FL

- Computer Vision focus
- Math minor (Linear Algebra, Partial Differential Equations)

STRENGTHS

- Algorithm/Data-structure design and analysis
- Parsing complex systems and code-bases
- Open Source Software/Hardware
- Linux application development

TOOLS

C++ C Python Java Golang

Intel SGX

Kubernetes Pulumi Docker Podman

Ostree

Linux Fedora ArchLinux

GDB Ghidra CMake GoogleTest

OpenCV Protobuf Messagepack

OTHER PROJECTS

Automated development lab deployment

📅 Jul 2023 – Present

Automated deployment of self-hosted lab environment and private VPN.

- Server running Fedora CoreOS image, pre-built via Containerfiles, and deployed via IPXE.
 - Kubernetes nodes running in Incus/LXD containers
 - Certificates from Kubernetes nodes updated and collected via Ansible script run after deployment
 - Automated deployment of Headscale control server to fly.io using Pulumi and terraform-fly
 - Deployment and configuration of services (Authentik, Redis, Postgres, git) to Kubernetes using Pulumi
 - Remote access to local services via Headscale + Router (radvd + dnsmasq in Incus container)
-

Scene Reconstruction from Stereograms

📅 Aug 2021 – Apr 2022

A 1-year group research project for the UCF Center for Humanites and Digital Research, aiming to extract measurements from a historical stereogram.

- Detailed review of the algebra of camera projection, the state-of-the-art in 3D reconstruction, and how warping and other unknowns in our physical photographs might affect the process.
 - Designed a GUI application using PyQt5 allowing users to input a pair of cropped stereograms and then measure the distances between 2 points in the scene
 - Used C++ and several computer vision libraries to implement a stereogram processing pipeline
 - OpenCV: ORB feature matching, initial camera pose estimation, and image rectification
 - GTSAM: camera pose refinement using bundle adjustment
 - LocalExpStereo: generates per-pixel depth/disparity information from rectified images
 - Python bindings generated using pybind11
-

constyaml: C++20 compile-time YAML parser

📅 Apr 2022 – May 2023

📍 <https://github.com/carbonxiii/constyaml>

A project demonstrating how C++20's expanded constexpr support allows implementation of complex algorithms at compile time.

- YAML parsing following the strict YAML specification
 - Allows fully parsing configurations at compile time, making it a useful replacement for preprocessor definitions
 - Parsing results in a nested tuple-like type, minimizing extra space in the compiled binary
 - Includes several constexpr data-structures reference implementations
 - Built around a constexpr state machine
-

Harness: Simple Raspberry Pi KVM

📅 May 2022 – Sep 2022

📍 <https://github.com/carbonxiii/harness>

- Cheap KVM alternative using a Raspberry Pi ZeroW to spoof a USB keyboard and touchpad, and an HDMI capture card
- Allows a keyboard/mouse to be directly connected to the host, then relayed to a machine where the "harness" is plugged in
- A capture card is used to mirror the harnessed machine's display on the host
- Mouse events are relayed as absolute coordinates, mapping from the window coordinates to the harnessed machine's display coordinates
- Cross-compiles and generates a trimmed image for the Pi using Packer and Docker