

ERICK ALANIS

+1 (720) 809-0857 | eafc677698@gmail.com | linkedin.com/in/alanis-erick | github.com/EA677698 | www.erickalanis.com/

EDUCATION

University of Colorado - Boulder

August 2020 - December 2024

Bachelor's, Computer Science

GPA: 3.013

- Relevant coursework: Operating Systems, Network Systems, Linux System Administration, Ethical Hacking, Compiler Construction, Computer Organization.

PROFESSIONAL EXPERIENCE

NVIDIA

Santa Clara, CA, USA

Embedded Systems Intern

May 2024 - August 2024

- Engineered a cross-platform containerized solution for NVIDIA's L4T (Linux for Tegra) flashing tool using Docker, enabling support beyond Ubuntu Linux, including Windows via WSL2. This required extensive research into USB passthrough, Linux device trees, and Docker's platform-specific behaviors.
- Prototyped a native WSL2 solution leveraging USB over TCP to bypass Docker entirely, successfully enabling device communication within WSL2 environments.
- Investigated and resolved bugs within the C-based flashing tool, enhancing developer experience by fixing inconsistent messaging and improving debugging visibility.
- Developed a full-stack visualization dashboard using Streamlit, MongoDB, and Jenkins, displaying OS image test results via interactive graphs. Designed filters by date and metadata to support QA engineers in monitoring regressions and analyzing test history.

PROJECTS & OUTSIDE EXPERIENCE

Collegiate Penetration Testing Competition

San Jose, CA, USA

Team Member

November 2023 - November 2023

- Executed a simulated penetration test as part of a six-member consulting team, performing a comprehensive assessment of a segmented enterprise network to identify and report security vulnerabilities.
- Used tools from the Kali Linux distribution — including Nmap, Netcat, and Hydra — to enumerate 4 separate subnets and scan over 1,000 IP addresses, identifying open ports and potential attack vectors.

RemiNEScent

July 2023 - Present

A Nintendo Entertainment System emulator

- Implemented all 56 official and 256 unofficial 6502 CPU instructions, along with their various addressing modes, ensuring emulation accuracy.
- Established a continuous integration pipeline using GitHub Actions and Pytest, incorporating community test ROMs to enhance reliability and streamline regression testing.
- Utilized bit-wise operations for tasks such as memory addressing, memory mapping, and I/O port control for accuracy.

RISC-V PIPELINED CPU

August 2024 - December 2024

A 32-bit, 5-stage pipelined CPU using the RISC-V instruction set.

- Designed and implemented a 5-stage pipelined CPU (Fetch, Decode, Execute, Memory, Writeback) using Cudasip resulting in an accurate simulation of the designed CPU.
- Added mechanisms for hazard detection and resolution and error correction, minimizing pipeline stalls and improving reliability.
- Implemented all base RISC-V instructions (RV32I), including branching, arithmetic, and load/store operations alongside two custom ECC encoding and decoding instructions.

SKILLS

Relevant Programming Software and Hardware: Node.js, Bootstrap, Linux/Unix, MySQL, SQL, Express.js, TCP/IP, Git, Containers, Docker, CLion, IntelliJ, VSCode, Arduino IDE, GDB Debugger, Confluence, GitHub, GitLab, Jenkins, Pytest, Windows Subsystem for Linux, ESP32, MongoDB, PostgreSQL, VSCode, VirtualBox, VMWare, Git, Windows OS, MacOS, Multiple Linux Distributions, Raspberry Pi, SSH, Telnet, DHCP, ARP

Programming Languages: Java, C/C++, Bash, JavaScript, Python, x86-64, RISC-V, HTML/CSS, C#, ARM, Rust

Languages: Spanish, English