

# DIEGO WEARDEN

diegowearden@utexas.edu • (210) 990-1792 • Austin, TX  
github.com/DiegoWearden • linkedin.com/in/diegowearden

## EDUCATION

---

<b>The University of Texas at Austin</b> , Austin, TX	<i>Bachelor of Science, Computer Science</i>	May 2026
	GPA 3.63/4.0	
	Minor in Statistics and Data Science	

Relevant Coursework: Multicore Operating Systems, Computer Architecture, Compilers, Computer Security, Data Structures, Algorithms

## SKILLS

**Security & Systems:** Memory corruption exploitation (stack smashing, heap exploitation, heap grooming, format string vulnerabilities), Control-Flow Integrity, Return-Oriented Programming, Software Fault Isolation, Secure Software Development, MMU configuration, Process scheduling, File system management, Interrupt handling & Version Control

**Technical/Computer Skills:** Expert in Java; Experienced in C & C++; Expert in Linux; Expert using GDB; Experienced in Operating System Development; Experienced in ARM & x86 Assembly; Experienced in Shell Scripting; Experienced in Git & Version Control

**Languages:** Fluent in English and Spanish

## EXPERIENCE

---

**Undergraduate Researcher - [Learning-Directed OS \(LDOS\)](#)** - C, Assembly, Bash, Python Fall 2024-Present

- Collaborated with researchers across 4 universities to develop a self-adaptive operating system using machine learning to maximize performance
- Assessed scheduler behavior under adversarial workloads to evaluate robustness
- Collected 10,000,000+ data points on latency to evaluate the performance of 6 different linux schedulers
- Revealed 30% more variable latency in ML vs. CFS under mixed workloads.
- Automated kernel logging with Bash and Python for reproducible performance benchmarks on workloads

## PROJECTS

---

**Operating System Development** - C, C++ Spring 2024-Present

- Designed a distributed OS for ARM hardware and a bare-metal Raspberry Pi 3 Model B, with device drivers and an ELF loader, using QEMU for testing
- Experience in developing in constrained bare metal environments
- Tested complex components such as context switching, interrupt handling, and file system management (Ext4)
- Gained proficiency in Linux kernel development and debugging low-level concurrent system issues

**Computer Architecture Projects** - C, Assembly Fall 2024

- Implemented an ARM pipelined processor to simulate fetch, decode, execution, memory access, and write-back stages, enabling the execution of machine code from ARM assembly instructions
- Developed a cache hierarchy model, to optimize data access times and improve CPU performance on ARM
- Worked with assembly language and hardware description languages to understand instruction set architectures across multiple platforms, including ARM, x86, and RISC-V

**Computer Security Projects** - C, RISC-V Assembly, Javascript Fall 2024

- Exploited stack-based buffer overflows, heap exploitation, and format string attacks, to hijack program execution
- Exploited QuickJS type confusion vulnerabilities to create fake objects and execute arbitrary code
- Bypassed bounds-check elimination on IonMonkey range analysis of Object.keys().length
- Designed an encoder that transforms 64-bit RISC-V payloads into shellcode free of NUL bytes and valid as UTF-8
- Implemented and broke out of trusted execution environments using software fault isolation

---

Work Eligibility: Eligible to work in the U.S. with no restrictions