SUNNY LAN

sunny.lan@uwaterloo.ca
github.com/sunny-lan
sunny-lan.github.io/

WORK EXPERIENCE

Core Gfx - Kernel Mode Driver Team

May 2020 - August 2020

AMD - Markham, ON

- Mainly worked in C/C++, occasionally scripting in Python. Very large codebase (>1M lines of code)
- Mainly worked with the Hardware Scheduler team
- Helped reproduce, triage, and debug tickets (using WinDBG)
- Implemented basic tests for the hardware scheduler
- Assist with enabling and verifying page migration on multi GPU systems (Vega)
- Took ownership on Windows debugging tool for hardware scheduler

AMD - Markham, ON

January 2021 - May 2021

- Continue working with Hardware Scheduler team, maintaining previous projects (such as debugger)
- Mainly worked on new tool with mentor, the Firmware Profiler (see projects section)
- Add code to collect additional telemetry data in the driver and in the hardware scheduler firmware

AWARDS

CCO Silver (May 2019) and CCO Bronze (May 2018)

- National programming competition, with problems involving topics such as graph theory, data structures, and dynamic programming
- Ranked Top 15 in Canada, out of 2700 contestants

PicoCTF 2nd place in Canada (November 2018)

- Cybersecurity competition, with challenges involving topics such as binary exploitation and reverse engineering, as well as web security. Knowledge of Linux systems and x86/64 assembly is required to successfully complete the challenges.

PROJECTS

E-Wall

September - December 2019

Python, OpenCV, Pygame

- A set of games where characters can interact with reality (such as by standing on your hand)
- Used OpenCV's Sobel edge detection and FLANN matching to detect physical objects
- Built physics engine capable of handling rapidly changing collision data from webcam
- Image processing is multi-threaded and physics work is mostly done in native code

Firmware Profiler

January 2021 - May 2021

Rust, C++, Assembly

- Tool meant to measure performance overhead incurred by GPU firmware without requiring real hardware (runs on Windows/Linux). It allows user to give custom program which runs simultaneously with firmware and can interact with it (sending commands/interrupts etc.)
- Took ownership and implemented majority of project, including basic OS functions such as memory management, paging, thread scheduling, multicore mode, as well as the firmware communication layer, and per-function performance measurement

EDUCATION

Candidate for Bachelor of Software Engineering, University of Waterloo (2019 - present)