Teaching Operating Systems: A Ground Up Approach

Maximilian Sonderegger
College of Information and Computer Sciences
msonderegger@umass.edu
12/11/2022

Advisor: Tim Richards
Second Committee Member: Joe Chiu
Research Type: Project

1. Goal

Teaching operating systems is hard. Understanding high level abstractions like virtual memory, processes, and file systems is no longer sufficient when answering questions like “How do the memory management unit and translation look aside buffer allow CPUs to implement virtual memory?”, “How do you switch processes?”, and “How do you get data off of a disk?”. However, understanding the answers to these questions requires a deep understanding of complex topics including assembly, CPU architectures, and device drivers. Current operating systems curricula avoid this complexity by focusing on a top-down view of operating systems and dissecting how their implementations work in concert to provide all the features expected in a modern computer.

Although this approach does a good job of introducing students to the problems operating system designers face and the strategies they use to solve them, it does not do a good job of teaching students the actual methods used to write operating systems. This approach ultimately fails to engage students because the interesting part of making an operating system is designing the system as a whole, not implementing a particular scheduling algorithm. Despite this, it is a common approach because of the immense complexity of modern operating systems (Linux has over 8 million lines of code). However, there are other alternatives. The success of MIT’s Xv6 operating system, shows that a functioning operating system can be written with a minimal amount code, opening the possibility for other approaches.

I aim to explore the possibility of a class where students learn about operating systems from the ground up, starting with a bare metal CPU. I believe that students will be more engaged if they are building their own operating system and will retain what they have learned longer by drawing on their own experience in OS design. Professor Richards’ seminal 377H colloquium has shown that students are eager to learn operating systems in this fundamental way. I plan to work with Professor Richards and and Professor Chiu to design a curriculum for a one semester course where students implement an operating system similar to Xv6 over the course of the semester through weekly projects, in order to understand the feasibility of such a class. Additionally, I plan to research the possibility of using Rust as an educational language and replacement for the C programming language which xv6 uses for its implementation.

2. Key Readings

- “The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Document Version 20191213”, Editors Andrew Waterman and Krste Asanovi´ c, RISC-V Foundation, December 2019. Retrieved December 12, 2022, from https://github.com/riscv/riscv-isa-manual/releases/download/Ratified-IMAFDQC/riscv-spec-20191213.pdf.

- “The RISC-V Instruction Set Manual, Volume II: Privileged Architecture, Document Version 20211203”, Editors Andrew Waterman, Krste Asanovi´ c, and John Hauser, RISC-V International, December 2021. Retrieved December 12, 2022, from https://github.com/riscv/riscv-isa-manual/releases/download/Priv-v1.12/riscv-privileged-20211203.pdf.

- “xv6: a simple, Unix-like teaching operating system”, Russ Cox, Frans Kaashoek, and Robert Morris, MIT, August 2021. Retrieved December 12, 2022, from https://pdos.csail.mit.edu/6.S081/2020/xv6/book-riscv-rev1.pdf.

- “Safe Systems Programming in Rust”, Ralf Jung, et al., Communications of the ACM, April 2021. Retrieved December 12, 2022, from https://dl.acm.org/doi/10.1145/3418295.

- “Teaching Operating Systems: Just Enough Abstraction”, Philip Machanik, Rhodes University, July 2016. Retrieved December 12, 2022, from http://dx.doi.org/10.1007/978-3-319-47680-3_10.

3. Communication Plan

I plan to check in with Professor Richards weekly for approximately 30-60 minutes, in person and over email, to discuss my current progress. He has discussed the possibility of forming a systems lab on campus, which would provide the perfect opportunity to have regular meetings with him and with other students who are working on similar projects. I can use this time to inform him of what I am working on currently, what I am struggling with, and how I am doing on my semester long goals. He will be able to offer me feedback, and help me work towards solutions, as necessary.

4. Specialized Training

None required.

5. Method or Agenda

My focus will be on designing a small operating system for the RISCV instruction set that I can use to teach a course on operating system design. To do this I will use QEMU, a popular and commonly available CPU emulator often used for operating system implementation and exploration. I plan to write the OS using a combination of assembly, linker scripts, and C (or possibly Rust). To do this I will use *ld* and the gcc-riscv toolchain (or the Rust toolchain). Evaluating a curriculum requires more experience than I have, so I plan to rely on the expertise of my two advisors to quantify my work. As part of designing a curriculum I will also design scripts that can grade submissions using the Gradescope auto grader. I will evaluate different ways to create the development environment required for this class, ensuring that all students have access to the necessary tools. Because my overall goal is for students to be able to easily understand the operating system over the course of a semester, I will need a way to measure the simplicity of my operating system. I plan to use significant lines of code (SLOC) as my primary metric for evaluating this, as in general the less code there is the easier it is to understand the whole thing.

6. Timeline

Dates are in DD/MM/YYYY

- 17/02/2022: Written summary of reading 1
- 03/03/2022: Written summary of reading 2
- 17/03/2022: Written summary of reading 3
- 24/03/2022: Written summary of reading 4
- 31/03/2022: Written summary of reading 5
- 14/04/2022: Completion of literature review writeup
- 28/04/2022: First draft of project proposal on CHC paths
- 12/05/2022: Final submission of project proposal on CHC paths