CS 380L Advanced Operating Systems Spring 2023

Course Objectives

This course considers advanced operating system topics and exposes students to recent developments in operating systems research. The course involves readings and lectures on classic and recent papers on a range of topics, including OS design, virtual memory management, virtualization, concurrency and synchronization, file systems, cloud systems, heterogeneity, and security. The course also exposes students to basic system-building and evaluation methodologies through a handful of programming assignments and a final project.

The course assumes background commensurate with that provided by an undergraduate course on operating systems such as UT's CS 439. If you have not had a formal OS course, please discuss with the instructor before committing to the course.

The course objectives include:

- Reading classic systems papers that shaped the field.
- Understanding systems concepts like virtualization.
- Gaining practical experience with systems programming, tools, and experimentation.
- Gaining experience with defining and refining a research project.
- Presenting technical materials to others orally and in writing.

Teaching Assistant

Samuel Laberge	`samuel DOT laberge AT utexas DOT edu`	Tuesdays 2-3pm at desk 1 in the GDC basement	zoom and in person at TBA

Instructor

Emmett Witchel	`witchel AT cs DOT utexas DOT edu`	GDC 6.432	Zoom Wednesdays 3:30pm-4:30pm, or by appointment

Course Materials

Readings. There is no textbook for this course. The course is based on a collection of journal, conference, and other papers that describe the history and state of the art in operating systems. The preliminary list of papers and schedule is available here. You must read the papers before class. At a minimum we recommend two close readings. We will provide papers online from a machine **in the utexas.edu domain**, but the majority should be available on the web.

For background reading about Linux, I recommend the three books below. While I will likely refer to materials from these books in class, they are not required. That said, these books are all very much a worthwhile investment and great references for anyone doing systems research.

- Understanding the Linux Kernel (3rd Edition) by Daniel P. Bovet and Marco Cesati
- Linux Kernel Development (3rd Edition) by Robert Love

- Professional Linux Kernel Architecture by Wolfgang Mauerer.

Online resources. I have created CS 380L Witchel in piazza (piazza.com) for this course (here). I will use it mostly for announcements and as a discussion forum, but feel free to provide feedback, and suggestions. Similarly, feel free to use email for questions and discussion.

Class Participation

Class participation is vital to the success of the course. Class time will not be used to rehash the material in the papers, and the instructor will avoid lecturing directly on the material. Rather, class time will be devoted highlighting important ideas and promoting deeper discussion about interesting features. The reading load is heavy, and depending on your level of experience reading technical papers, the load can be as high as 10-15 hours of reading per week. *So...only take this course if you are willing and able to do a lot of reading.*

If you cannot do the reading for a particular class, please contact me. In general, I would prefer students who haven't done the reading do not come to class. **You are allowed two absences but you must consult me if there will be more.** I will ask direct questions about the paper and will expect you to be able to answer, and in-class quizzes should be expected. If a paper is too confusing to understand, please write a critique (according to guidelines below) to demonstrate your effort to engage, and hopefully encourage discussion from others who struggled similarly.

Paper critiques

A template review form for papers is here. I strongly encourage you to actually write out a review/reaction for each of these papers. Not only will it help you understand your own thoughts, but it helps you solidify your understanding, and provides you with an artifact to which you can refer to help refresh your memory in the future. I do not **plan** to require that these reviews be handed in, however, I reserve the right to require a handful, depending on how well prepared the class at large seems to be for discussions.

In-class Presentation

The course includes a final project, which can be done individually or in pairs. During one of the final class meetings, each project group is responsible for a presentation about their project.

Programming Assignments

The course requires several programming assignments that will give you experience in building, booting and running an operating system. The assignments will also expose you to methodological systems issues such as how to model, measure and report performance, how to design a workload to test kernel functionality, and the dependence on workload for the evaluation of a system feature. Finally the assignments will expose you to how to write about systems, their design, implementation, and measurement.

The goal of the assignments is to demystify the operating system and convince you that the OS really is just another (super-interesting) program. Sometimes puzzling system behavior can be understood and worked around by reading and understanding the source code of the OS. Why did mmap return ENOMEM? There are several distinct possibilities that you can see in the code. These

assignments might even give you a bit of practical knowledge, for example allowing you to get Linux to recognize your fancy, new USB device.

We will use Linux and the KVM virtual machine. Unfortunately, the CS machines are not set up to allow use of KVM. The easiest solution is for you to find a machine that runs Linux on which you have root privilege. Like a laptop. Failing that, I will give out accounts on my group's machines for you to run experiments.

Grading

Your final grade for the course will be based on the following approximate weights. Canvas will have more precise breakdowns.

- 20-35% Class participation, in-class quizzes, your presentation and homeworks/labs
- 35-60% Two exams
- 20-40% Project

Course Policies

Collaboration

- Students are encouraged to pair up for projects
- All exams are done individually, with absolutely no collaboration.
- Each student must present twice, though partners can collaborate on a project presentation
- I strongly encourage you to discuss the papers and the homeworks with anyone you can. That's the way good science happens. But all work and writeup for the homework must be your own, and only your own.
- As a professional, you should acknowledge significant contributions or collaborations in your written or spoken presentations.

  Students who violate University rules on academic misconduct are subject to the student conduct process and potential disciplinary action. A student found responsible for academic misconduct may be assigned both a status sanction and a grade impact for the course. The grade impact could range from a zero on the assignment in question up to a failing grade in the course. A status sanction can range from probation, deferred suspension and/or dismissal from the University. To learn more about academic integrity standards, tips for avoiding a potential academic misconduct violation, and the overall conduct process, please visit the Student Conduct and Academic Integrity website at:
  http://deanofstudents.utexas.edu/conduct.

  The student code of conduct is here. Intellectual dishonesty can end your career, and it is your responsibility to stay on the right side of the line. If you are not sure about something, ask.

Exams

- You will at least be allowed a page of notes during exams. I will likely make the exams open book, but I reserve the right not to do so.
- Laptops, tablets, and ereaders are BANNED from exams. You should not need them in an exam, and they are far too flexible as communication devices to make enforcement of non-

communication policies enforceable. Any use of a communication device for any reason in the exam room will earn an automatic zero on the exam.

Special offer: you can write your own exam questions! Submit a question with your solution in advance of the exam, and if we like it, it will appear on the exam.

Late Policy

- All labs must be submitted in class the day they are due.
- Late labs will be graded at the discretion of the instructor.
- In this class, it is always better to do the work (even late) than not do it at all.
- *If you become ill*: Contact the instructor and get a medical note. A medical note is required to miss an exam.

Last updated: 2023-08-17 23:38:39 -0500 [validate xhtml]