



CSCI 555L/655: Advanced Operating Systems

Units: 4.0

Spring 2024 MW 2:00-3:50pm

Location: VPD 112

Instructor: Harsha V. Madhyastha

Office: SAL 326

Office Hours: TBD

Contact Info: Post privately on Piazza to contact the instructor

Course Description

The process of developing a new software system – a cloud-hosted web service, a multi-threaded smartphone application, a key-value store, etc. – rarely involves invention of new techniques. Instead, any system design is typically based on a careful combination of well-known techniques, taking into account the constraints and objectives of the system’s target workload and the environment in which the system will be developed. In this course, first-year graduate students will learn the common techniques used to make a system reliable and performant at scale. In every lecture, we will discuss 1-2 research papers, distill the techniques used in those systems, understand the authors’ motivations in choosing to use those techniques, and discuss the applicability of those techniques in other settings. In groups of 2, every student will lead the discussion for one of these papers over the course of the semester. Students will also conduct a semester-long project in which they will implement and evaluate a prototype system in groups of 2-3.

Learning Objectives

By the end of this course, students will have gained experience with

- techniques for dealing with failures caused by concurrency bugs, power outages, hardware corruption, etc.
- techniques for optimizing latency and throughput in a variety of settings
- techniques for enabling a system to scale to a large number of users, data items, computers, etc.
- reading and distilling a research paper in the area of computer systems
- presenting the findings of a research paper in the area of computer systems
- implementing some of the techniques learned in an example prototype system

Recommended Preparation

Students are expected to have completed an undergraduate course on operating systems, and have a basic understanding of concurrency using threads, virtual memory management using paging and segmentation, and data storage using file systems.

Course Notes

Students enrolled in this course will receive a letter grade.

All lectures will be recorded to enable students to review the lectures, but not to use as a substitute for attending lecture.

Lecture slides and recordings will be posted on a website dedicated for this course.

Piazza will be used for announcements and course discussions.

Readings and Supplementary Materials

There is no textbook for this course.

For every lecture, 1 or 2 research papers will be posted on the course website, which students are required to read before coming to class.

Project

Students will work in groups of two or three on a semester-long project of their choice. There will be four milestones:

- Week 3: Form a group and pick a topic for your project
- Week 6: Submit two-page project proposal
- Week 10: Submit five-page intermediate project report
- Week 15: Present project outcomes in class and submit ten-page final project report

The project counts for 40% of your final grade.

To help students pick a project, at the start of the semester, the instructor will discuss strategies for choosing a project topic. Examples include a measurement-based characterization of a deployed or open-source system, the development of a new software system, or reproducing results from the evaluation section of a published paper.

Project Groups

Declare the membership of your project group by the third week of the semester. After this date, we will form groups from the remaining students. Choose your group members carefully. You should discuss topics such as prior experience, course background, goals for this course, workload and schedule for this semester, and preferred project management and work style. Make sure you can find several blocks of time during the week to meet to discuss or carry out the project.

Students are expected to work diligently in their group for the benefit of the entire group. All group members should be familiar with all aspects of each project, irrespective of their role on the project. We expect all group members to contribute their fair share, and we expect to assign the same project grade to all members of a group. **Members who either contribute less than their share or prevent others from contributing their share may receive a lower grade on the project; non-contributing members will receive a zero.** In case of disputes regarding contribution, the instructor may interview group members and inspect commits to any code repository used by the group. Commits to the repository should reflect the proportion of work performed by each group member. If you use pair programming, take turns at the keyboard so that the commit log reflects the contributions of all members.

Students may be fired from a group by the remaining members. The procedure for this is as follows: (1) documented "gentle warning" of risk of firing via both e-mail and **private** Piazza post (with instructor included on both), with cause and specific work required to remain in group; (2) allow at least 72 hours for compliance; (3) if the problems persist, e-mail statement of firing to the group and post **privately** to the instructors via Piazza. Fired group members may join another group; students who cannot find a group must complete a project by themselves.

Managing group dynamics and using each group member's time and talents effectively can be difficult. If there are problems with your group, please see an instructor as soon as possible. Be open and candid with your group about potential problems early on so your group can plan around those problems and not fall behind. A sure way to make your group upset at you is not finishing your work at an agreed-upon deadline and not informing them about the problems early enough for them to help. Similarly, it is not okay to try to do all of your group's work on a project by yourself, rather than in collaboration with your group.

Avoid doing all the work on the project in group programming sessions. Usually someone on a team is faster than the others, and that student will figure out the solution to most problems. A better approach is to end every group session with a set of problems that need to be solved. Every team member can then outline the solutions to these problems individually on their own time. The group can collaboratively work out the details at the next meeting.

Quizzes

There will be three in-class quizzes over the course of the semester. There will be no midterm or final exam. The dates for all quizzes will be announced in the first lecture.

If you miss a quiz for reasons other than a documented medical or personal emergency, you will receive a zero for that quiz. Quiz dates are given at the beginning of the semester so you can avoid scheduling job interviews or other commitments on those days. Outside commitments are not considered a valid reason for missing a quiz.

Grading Timeline

Scores for every quiz will be released roughly a week after the day the quiz is conducted.

Classroom Norms

Lectures will make heavy use of in-class discussions. Students are expected to come to class having read the papers assigned for the class and actively participate in in-class discussions. **We ask that all students treat each other with respect.**

Attendance and Participation

Students are expected to attend every lecture in-person. Class participation will count towards 20% of a student's grade.

In pairs of two, every student will be in charge of leading the in-class discussion of one paper over the course of the semester. The instructor will work with students to help them prepare for their in-class presentation. This will count for 10% of every student's final grade.

Reading List

The following is a tentative list of the papers that we will discuss over the course of the semester.

How to read and evaluate papers?

- How to Read a Paper ([pdf](#))
- An Evaluation of the Ninth SOSP Submissions or How (and How Not) to Write a Good Systems Paper ([pdf](#))

Reliability

- RAID: High-Performance, Reliable Secondary Storage ([pdf](#))
- Disconnected operation in the Coda file system ([pdf](#))
- An Analysis of Data Corruption in the Storage Stack ([pdf](#))
- The Episode File System ([pdf](#))
- Reimplementing the Cedar File System Using Logging and Group Commit ([pdf](#))
- Cores that don't count ([pdf](#))
- Eraser: A Dynamic Data Race Detector for Multithreaded Programs ([pdf](#))
- Improving the Reliability of Commodity Operating Systems ([pdf](#))

Performance

- Caching in the Sprite Network File System ([pdf](#))
- Speculative Execution in a Distributed File System ([pdf](#))

- Outatime: Using Speculation to Enable Low-Latency Continuous Interaction for Mobile Cloud Gaming ([pdf](#))
- Rethink the sync ([pdf](#))
- The design and implementation of a log-structured file system ([pdf](#))
- Better I/O Through Byte-Addressable, Persistent Memory ([pdf](#))
- Coz: Finding Code that Counts with Causal Profiling ([pdf](#))
- Horcrux: Automatic JavaScript Parallelism for Resource-Efficient Web Computation ([pdf](#))

Other considerations

- Difference engine: Harnessing memory redundancy in virtual machines ([pdf](#))
- A study of practical deduplication ([pdf](#))
- The multikernel: A new OS architecture for scalable multicore systems ([pdf](#))
- Energy-aware adaptation for mobile applications ([pdf](#))
- Shielding Applications from an Untrusted Cloud with Haven ([pdf](#))

Distributed Systems

- Implementing fault-tolerant services using the state machine approach: A tutorial ([pdf](#))
- Improving MapReduce performance in heterogeneous environments ([pdf](#))
- Cluster-based scalable network services ([pdf](#))
- Efficient Memory Disaggregation with Infiniswap ([pdf](#))
- Chord: A scalable peer-to-peer lookup service for internet applications ([pdf](#))

Lessons

- Hints for computer system design ([pdf](#))
- On system design ([pdf](#))

Grading Breakdown

Table 1 Grading Breakdown

Assessment Tool (assignments)	Points	% of Grade
Project		40%
Quizzes		30%
Paper presentation		10%
Class participation		20%
TOTAL		100%

Final letter grades will be determined using a "modified" curve. We will assign grades of C and BELOW to individuals who do not perform satisfactorily (i.e., you should not assume that because this is a graduate class you will get a B or even B- if you perform unsatisfactorily). We will not assign incompletes unless it is for a documented medical reason (in accordance with USC policy).

PhD students enrolled in CSCI 655 will be required to do some additional work compared to those enrolled in CSCI 555L. Specifically, they will be asked to submit a review of a research paper in the area of computer systems that is not discussed in class. This review will be considered as equivalent to a fourth quiz and account for a quarter of the 30% grade contribution assigned to quizzes.

Academic Integrity

The University of Southern California is foremost a learning community committed to fostering successful scholars and researchers dedicated to the pursuit of knowledge and the transmission of ideas. Academic misconduct is in contrast to the university's mission to educate students through a broad array of first-rank

academic, professional, and extracurricular programs and includes any act of dishonesty in the submission of academic work (either in draft or final form).

This course will follow the expectations for academic integrity as stated in the [USC Student Handbook](#). All students are expected to submit assignments that are original work and prepared specifically for the course/section in this academic term. You may not submit work written by others or “recycle” work prepared for other courses without obtaining written permission from the instructor(s). Students suspected of engaging in academic misconduct will be reported to the Office of Academic Integrity.

Other violations of academic misconduct include, but are not limited to, cheating, plagiarism, fabrication (e.g., falsifying data), knowingly assisting others in acts of academic dishonesty, and any act that gains or is intended to gain an unfair academic advantage.

The impact of academic dishonesty is far-reaching and is considered a serious offense against the university and could result in outcomes such as failure on the assignment, failure in the course, suspension, or even expulsion from the university.

For more information about academic integrity see the [student handbook](#) or the [Office of Academic Integrity's website](#), and university policies on [Research and Scholarship Misconduct](#).

We encourage students to help each other learn the course material. As in most courses, there is a boundary separating these two situations. You may give or receive help on concepts covered in lecture or discussion and on the specifics of Go syntax. You may consult with other students to help you understand the project specification (i.e., the problem definition). However, you may not collaborate in any way when constructing your solution; the solution to the project must be generated by you or your group working alone. Any misrepresentation of another person's work as your own is unacceptable and is a violation of the honor code. You are not allowed to work out the programming details of the problems with anyone or to collaborate to the extent that your programs are identifiably similar. You are not allowed to look at or in any way derive advantage from solutions or code that you did not write.

If you have any questions as to what constitutes unacceptable collaboration, please talk with the instructor. You are expected to take reasonable precautions to protect your work. You may not post your work in a publicly accessible location, such as public code repositories. Don't let other students borrow your account or computer; don't leave your program in a publicly accessible directory; and don't discard printouts in a public place.

If found responsible for an academic violation, students may be assigned university outcomes, such as suspension or expulsion from the university, and grade penalties, such as an “F” grade on the assignment, exam, and/or in the course.

Course Content Distribution and Synchronous Session Recordings Policies

USC has policies that prohibit recording and distribution of any synchronous and asynchronous course content outside of the learning environment.

Recording a university class without the express permission of the instructor and announcement to the class, or unless conducted pursuant to an Office of Student Accessibility Services (OSAS) accommodation. Recording can inhibit free discussion in the future, and thus infringe on the academic freedom of other students as well as the instructor. ([Living our Unifying Values: The USC Student Handbook](#), page 13).

Distribution or use of notes, recordings, exams, or other intellectual property, based on university classes or lectures without the express permission of the instructor for purposes other than individual or group study. This includes but is not limited to providing materials for distribution by services publishing course materials. This restriction on unauthorized use also applies to all information, which had been distributed to students or in any way had been displayed for use in relationship to the class, whether obtained in class, via

email, on the internet, or via any other media. ([Living our Unifying Values: The USC Student Handbook](#), page 13).

Students and Disability Accommodations

USC welcomes students with disabilities into all of the University's educational programs. The Office of Student Accessibility Services (OSAS) is responsible for the determination of appropriate accommodations for students who encounter disability-related barriers. Once a student has completed the OSAS process (registration, initial appointment, and submitted documentation) and accommodations are determined to be reasonable and appropriate, a Letter of Accommodation (LOA) will be available to generate for each course. The LOA must be given to each course instructor by the student and followed up with a discussion. This should be done as early in the semester as possible as accommodations are not retroactive. More information can be found at osas.usc.edu. You may contact OSAS at (213) 740-0776 or via email at osasfrontdesk@usc.edu.

Support Systems

[Counseling and Mental Health](#) - (213) 740-9355 – 24/7 on call

Free and confidential mental health treatment for students, including short-term psychotherapy, group counseling, stress fitness workshops, and crisis intervention.

[988 Suicide and Crisis Lifeline](#) - 988 for both calls and text messages – 24/7 on call

The 988 Suicide and Crisis Lifeline (formerly known as the National Suicide Prevention Lifeline) provides free and confidential emotional support to people in suicidal crisis or emotional distress 24 hours a day, 7 days a week, across the United States. The Lifeline is comprised of a national network of over 200 local crisis centers, combining custom local care and resources with national standards and best practices. The new, shorter phone number makes it easier for people to remember and access mental health crisis services (though the previous 1 (800) 273-8255 number will continue to function indefinitely) and represents a continued commitment to those in crisis.

[Relationship and Sexual Violence Prevention Services \(RSVP\)](#) - (213) 740-9355(WELL) – 24/7 on call

Free and confidential therapy services, workshops, and training for situations related to gender- and power-based harm (including sexual assault, intimate partner violence, and stalking).

[Office for Equity, Equal Opportunity, and Title IX \(EEO-TIX\)](#) - (213) 740-5086

Information about how to get help or help someone affected by harassment or discrimination, rights of protected classes, reporting options, and additional resources for students, faculty, staff, visitors, and applicants.

[Reporting Incidents of Bias or Harassment](#) - (213) 740-5086 or (213) 821-8298

Avenue to report incidents of bias, hate crimes, and microaggressions to the Office for Equity, Equal Opportunity, and Title for appropriate investigation, supportive measures, and response.

[The Office of Student Accessibility Services \(OSAS\)](#) - (213) 740-0776

OSAS ensures equal access for students with disabilities through providing academic accommodations and auxiliary aids in accordance with federal laws and university policy.

[USC Campus Support and Intervention](#) - (213) 740-0411

Assists students and families in resolving complex personal, financial, and academic issues adversely affecting their success as a student.

[Diversity, Equity and Inclusion](#) - (213) 740-2101

Information on events, programs and training, the Provost's Diversity and Inclusion Council, Diversity Liaisons for each academic school, chronology, participation, and various resources for students.

[USC Emergency](#) - UPC: (213) 740-4321, HSC: (323) 442-1000 – 24/7 on call

Emergency assistance and avenue to report a crime. Latest updates regarding safety, including ways in which instruction will be continued if an officially declared emergency makes travel to campus infeasible.

[USC Department of Public Safety](#) - UPC: (213) 740-6000, HSC: (323) 442-1200 – 24/7 on call

Non-emergency assistance or information.

[Office of the Ombuds](#) - (213) 821-9556 (UPC) / (323-442-0382 (HSC)

A safe and confidential place to share your USC-related issues with a University Ombuds who will work with you to explore options or paths to manage your concern.

[Occupational Therapy Faculty Practice](#) - (323) 442-2850 or otfp@med.usc.edu

Confidential Lifestyle Redesign services for USC students to support health promoting habits and routines that enhance quality of life and academic performance.