SJSU SAN JOSÉ STATE UNIVERSITY

College of Science · Computer Science

Operating Systems Section 04

CS 149

Fall 2024 3 Unit(s) 08/21/2024 to 12/09/2024 Modified 08/12/2024

Course Description and Requisites

Fundamentals: Contiguous and non-contiguous memory management; processor scheduling and interrupts; concurrent, mutually exclusive, synchronized and deadlocked processes; parallel computing;ï¿½files. Substantial programming project required.

Prerequisite(s): CS 47 or CMPE 102 (with a grade of "C-" or better), and CS 146 (with a grade of "C-" or better).ï¿½Allowed Declared Majors: Computer Science, Applied and Computational Math, Forensic Science: Digital Evidence, or Software Engineering Majors only; or Instructor Consent.

Letter Graded

Program Information

Diversity Statement - At SJSU, it is important to create a safe learning environment where we can explore, learn, and grow together. We strive to build a diverse, equitable, inclusive culture that values, encourages, and supports students from all backgrounds and experiences.

Course Learning Outcomes (CLOs)

I want you to learn skills that are useful beyond this class in future career. The following are concrete learning outcomes for this course.

1. High-level understanding of an Operating System and its role

2. Learn about the abstractions OS provides and their importance

3. Understand the inner workings of an OS and how they use modern hardware to work efficiency

4. Create correct multithreaded applications using synchronization primitives

5. Become (more) proficient in C

6. Understand how to navigate larg(er) codebases

Course Materials

Operating Systems: Three Easy Pieces (OSTEP) (http://pages.cs.wisc.edu/~remzi/OSTEP/) is the main textbook for this class. There is also a book on xv6: A simple, Unix like teaching OS (https://pdos.csail.mit.edu/6.828/2023/xv6/book-riscv-rev3.pdf), which we will use for the programming assignments.

Both books are free, and we will let you know which chapters to read as we move through the semester.

Course Requirements and Assignments

Programming Projects

Throughout the semester, we will give you programming assignments. Most, or all, of them will be based on xv6, an operating system that was specifically developed for educational purposes.

Note that all assignments **must be done by yourself.** Please do not ask other students for solutions or share your solution with other students.
It is fine to ask the instructor, TAs, or other students for clarification if you do not understand what the project description is asking you to do, but those conversations cannot be about any concrete implementation details. If you do receive help, please document it in your code.

Extensive help on a project will result in a reduced grade. Failure to document help, or any other forms of cheating, will result in a failing grade on the assignment at a minimum and may result in failure of the course. See Integrity (http://info.sjsu.edu/static/schedules/integrity.html)for more information.
Even in open source, you cannot copy code from one open source project to another without attribution.

We might also ask you to explain our solution to us, if we suspect you received help. In particular, we might do this if we think you generated the code with an LLM (or another AI tool), or if we think you used a solution that you found online.

Participation and Quizzes

Aside from exams and the programming assignments, a portion of your grade will also be based on (easy) homework quizzes and participation.

The quizzes will be on canvas and you will have multiple attempts. My goal here is not for you to have a hard time but a simple way to verify that you understood what we covered in class.

Participation really just means that you come to class and occasionally ask or answer a question. It does not really matter if your answers are correct or your questions are "good" (there are no bad questions). I simply want you to engage in class.

✓ Grading Information

The grade is based on a combination of programming assignments, exams, and quizzes.

Breakdown

Programming Projects	40%
Midterm 1	15%
Midterm 2	15%
Final	20%
Quizzes and Class Participation	10%

University Policies

Per University Policy S16-9 (PDF) (http://www.sjsu.edu/senate/docs/S16-9.pdf), relevant university policy concerning all courses, such as student responsibilities, academic integrity, accommodations, dropping and adding, consent for recording of class, etc. and available student services (e.g. learning assistance, counseling, and other resources) are listed on the Syllabus Information (https://www.sjsu.edu/curriculum/courses/syllabus-info.php) web page. Make sure to visit this page to review and be aware of these university policies and resources.

Course Schedule

The following is only a tentative schedule for the class, because this is my first time teaching this class at SJSU.
There is some slack at the end to either have more time for review content throughout the semester or cover some advanced topics.

Week No	Date	Topic
1	8/19	No Class
	8/21	Intro
2	8/26	Processes

	8/28	Scheduling
3	9/2	No Class
	9/4	Multi-level Feedback Queues
4	9/9	Memory Management
	9/11	Memory Segmentation
5	9/16	Pages
	9/18	Pages: TLB
6	9/23	Multi-Level Paging
	9/25	Review
7	9/30	Midterm #1
	10/3	Threads
4	10/7	Locks, Condition Variables
	10/9	More Locks and CVs
5	10/14	Semaphores
	10/16	Concurrent Datastructures
6	10/21	Concurrency Bugs
	10/23	I/O and Disks
7	10/28	RAID
	10/30	Filesystems
8	11/4	The Fast Filesystem (FFS)
	11/6	Review
9	11/11	No Class
	11/13	Midterm #2
10	11/18	Log Structured File Systems
	11/20	Solid State Drives
11	11/25	Network File System
	11/27	TBD
12	12/2	No class
	12/4	TBD
13	12/9	TBD