Course Syllabus

Edit

Science/Computer Science

CS 149, Operating Systems, Section 3, FALL 2024

Course and Contact Information

Instructor:	Sriram Rao
Office Location:	
Telephone:	
Email:	sriram.rao@sjsu.edu
Office Hours:	Fridays 12-1p (**Zoom (https://sjsu.zoom.us/j/85411886372?pwd=GEsFcaTWOS5Z3UDZMlvyNZBSn6bCbb.1)**) and by appointment
Class Days/Time:	Tue/Thu: 10:30-11:45a
Classroom:	MH 225
Prerequisites:	CS 47 or CMPE 102 **(https://catalog.sjsu.edu/content.php?filter%5B27%5D=CS&filter%5B29%5D=149&cur_cat_oid=15&navoid=5382#tt2687)** (with a grade of “C-” or better), and CS 146 (with a grade of “C-” or better). Proficiency in C programming.

Grader and Labs Information

Grader	Thinh Bui
Email	thinh.bui@sjsu.edu
Lab Hours	
Lab Location	
Office Hours	

Course Description

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

Course Learning Outcomes (CLO)

Upon successful completion of this course, students will be able to:

1. Explain the difference between kernel and user space.
2. Explain virtual memory management and page tables.
3. Analyze CPU scheduling policies.
4. Use threads and work with the concurrency issues that arise from them.
5. Analyze and implement concurrent data structures.
6. Identify and reason about ethical issues surrounding various operating system concepts.

Required Texts/Readings

Textbook

There are required two textbooks for this course. Both are free!

1. **Operating Systems: Three Easy Pieces (OSTEP) (http://pages.cs.wisc.edu/~remzi/OSTEP/)**

2. **xv6: A simple, Unix like teaching OS** (https://pdos.csail.mit.edu/6.828/2023/xv6/book-riscv-rev3.pdf)

To get hands-on experience working with OS code, we will be implementing many of the ideas that we will study in class. This will require you to write C programs.

References for learning Unix and C

Here is an excellent (introductory level) **free UNIX tutorial** (https://info-ee.surrey.ac.uk/Teaching/Unix/)

Here are a few references that will help you learn C programming

- A **free tutorial** (https://www.cs.hmc.edu/~rhodes/courses/cs134/sp19/readings/pointers.pdf) on Pointers and Arrays in C
- A standard reference for the C language is **The C Programming Language** (https://www.amazon.com/Programming-Language-2nd-Brian-Kernighan/dp/0131103628/ref=sr_1_1?crid=1B92AMP1JB4W8&dib=eyJ2IjoiMSJ9.7MU6AwhWtIZGSokpR_0dhrgvEkfVcs1qK6h9jSW5Cgh1Y9U1JydmrpUkPL-oYmnTIZJGXvr1QEuVIaJ3vb60uwGPzhc0CRIY6Zn3Y7O_AW3IAakwL4GFSsRmb2c7xEJwkrtn7fVpfROqwblhb8FC5OWkIHYJQgUHgvsRKWCVmHwaoRPcvClxuCEo-PbZzV97eKUsuXTcRrij6mMy3oiayxeWdrf3q_dWw5lVbefaK5Y.QZEkdagku9SSShlxkEkqE_SB_XLEq7EMdAwCaOTxD9A&dib_tag=se&keywords=c+programming+lang
1) book by Brian Kernighan and Dennis Ritchie.

Reading

To get the most out of this class, ***prior to each lecture***, students are encouraged to read the relevant chapter from the OSTEP book. Additionally, look at the corresponding chapter in xv6 book, which has hyperlinks that point to the related source code.

Other technology requirements / equipment / material

Programming assignments will be a significant part of this course. We will build C programs and use run them using qemu on Linux (Ubuntu 22.04). Hence, access to a computer that runs Linux (specifically, Ubuntu 22.04 distro) will be required.

Course Requirements and Assignments

I do not grade on a curve. The exams and projects measure what you are expected to have learned. There aren't many opportunities for extra credit apart from potential bonus questions on exams.

Programming Project: We will be doing **individual** programming assignments**. Individual programming assignments are not group projects.** If students get help on assignments, even to resolve a seemingly trivial problem, it must be documented in the code with the name of the person rendering the help and a brief description of the help provided. 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.

Programming assignments will be distributed via Github. Please create a Github account (if you don't already have one). We will use Gradescope for assignment submission and grading.

Help on Labs: Thinh Bui (course grader) will have weekly **in-person** "lab hours" in (Location TBD) to help you with the labs. Please seek his help whenever needed.

The **University Policy S16-9** (http://www.sjsu.edu/senate/docs/S16-9.pdf) , Course Syllabi (http://www.sjsu.edu/senate/docs/S16-9.pdf) requires the following language to be included in the syllabus:

“Success in this course is based on the expectation that students will spend, for each unit of credit, a minimum of 45 hours over the length of the course (normally three hours per unit per week) for instruction, preparation/studying, or course related activities, including but not limited to internships, labs, and clinical practica. Other course structures will have equivalent workload expectations as described in the syllabus.”

Final Examination or Evaluation

This course will have a cumulative final exam given during exam week. Per **University schedule** (https://www.sjsu.edu/classes/final-exam-schedule/fall-2024.php) this is on Friday, Dec. 13 from 9:45a-12p.

There will be two in-class exams given in the semester (the last being the final exam).

Grading Information

Determination of Grades

Grades will be calculated based on the individual project grades, the two mid-semester exams, the final, discussion participation. Briefly, the weighted distribution is,

Programming Project	40%

Midterm 1	15%
Midterm 2	15%
Final	20%
Class Participation	10%

The grade distribution will be:

Percentage	**Grade**
98 and above	A+
92-97	A
90-91	A-
88-89	B+
82-87	B
80-81	B-
78-79	C+
72-77	C
70-71	C-
68-69	D+
62-67	D
60-61	D-
59 and below	F

Classroom Protocol

This is your class. Please ask questions. Please come prepared. Do not engage in activity that may distract other students.

I do not take attendance except for the first two classes. Students not attending either of the first two classes will be dropped to make room for students on the waiting list. Attempting to get marked as present (by have someone else attend in your place or using technological deceptions) will be considered academic dishonesty and at a minimum will result in you getting dropped from the course.

This is my first time teaching this class. The schedule listed below is tentative and may change based on student needs. Furthermore, the programming project is new and we will learn by doing :-).

Class Participation

We will have weekly short quizzes in-class that covers material from the C/Unix tutorials linked above. These will complement material covered in class.

University Policies

Per University Policy S16-9, university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc. will be available on Office of Graduate and Undergraduate Programs' **Syllabus Information web page (http://www.sjsu.edu/gup/syllabusinfo/)** at http://www.sjsu.edu/gup/syllabusinfo/" Make sure to review these policies and resources.

Programming Project

This programming project comprises of **individual** programming assignments**. Individual programming assignments are not group projects.** The programs you submit must be your work and your work alone.

Caveats:

1. A requirement is that you **do not** publish solutions on public GitHub repos.
2. The labs are known to be demanding and debugging can be time consuming. So, please get started early for each lab.

Early submission:

- Submissions that are at least 2 or more days ***earlier*** than the deadline will get a 10% bonus.

Late Submissions:

1. Submissions that are up to a week late will be penalized 10% of the score.
2. Submissions that are up to two weeks late will be penalized 50% of the score.
3. I will not accept submissions that are more than two weeks beyond the submission date. This will be considered as a non-submission.
4. At my discretion, non-submissions will likely result in a grade penalty.

Cheating/Academic Dishonesty

I take issues of Academic Dishonesty very seriously. ***Do not cheat***. ***Do not share code***. If we detect cheating in a programming assignment, you will get a **0** for that lab. Repeat offense will likely lead to a **F** grade in the course.

Artificial intelligence (AI) tools like ChatGPT, Google Gemini, and GitHub Copilot are not permitted to be used as a replacement for the writing or problem-solving components of this class. SJSU's subscription to Turnitin has an AI-detection feature, and assignments that have been determined by that application or by other convincing evidence to have been written by AI in substantial fractions will receive an automatic zero. The incident will also be reported to the University as academic misconduct.

Course Schedule

(Tentative) Course Schedule

Week	Date	Topic/Theme	Reading(s)
		Pre-work: Learn UNIX command line tools	**UNIX Tutorial (https://info-ee.surrey.ac.uk/Teaching/Unix/)**
		Pre-read	**C tutorial (https://www.cs.hmc.edu/~rhodes/courses/cs134/sp19/readings/pointers.pdf)**
		Part I: Processes	
1	8/22	Introduction to OS	**OSTEP: Chapter 2 (http://pages.cs.wisc.edu/~remzi/OSTEP/intro.pdf)**
2	8/27	Process Abstraction	• **OSTEP: Chapter 4 (http://pages.cs.wisc.edu/~remzi/OSTEP/cpu-intro.pdf)** • **xv6: Chapter 1 (https://pdos.csail.mit.edu/6.828/2023/xv6/book-riscv-rev3.pdf)**
	8/29	Process management APIs	• **OSTEP: Chapter 5 (http://pages.cs.wisc.edu/~remzi/OSTEP/cpu-api.pdf)** • **xv6: Chapter 4 (https://pdos.csail.mit.edu/6.828/2023/xv6/book-riscv-rev3.pdf)**
3	9/3	Process Execution	**OSTEP: Chapter 6 (http://pages.cs.wisc.edu/~remzi/OSTEP/cpu-mechanisms.pdf)**
	9/5	Scheduling	• **OSTEP: Chapter 7 (http://pages.cs.wisc.edu/~remzi/OSTEP/cpu-sched.pdf)** • **OSTEP: Chapter 8 (https://pages.cs.wisc.edu/~remzi/OSTEP/cpu-sched-mlfq.pdf)**
4	9/10	Lottery Scheduling	**OSTEP: Chapter 9 (https://pages.cs.wisc.edu/~remzi/OSTEP/cpu-sched-lottery.pdf)**

		Part II: Memory	
	9/12	Intro to Virtual Memory	• OSTEP: Chapter 13 (http://pages.cs.wisc.edu/~remzi/OSTEP/vm-intro.pdf) , • OSTEP: Chapter 14 (http://pages.cs.wisc.edu/~remzi/OSTEP/vm-api.pdf)
5	9/17	Address Translation	OSTEP: Chapter 15 (http://pages.cs.wisc.edu/~remzi/OSTEP/vm-mechanism.pdf)
	9/19	**Midterm I**	
6	9/24	Segmentation	OSTEP: Chapter 16 (https://pages.cs.wisc.edu/~remzi/OSTEP/vm-segmentation.pdf)
	9/26	Paging	OSTEP: Chapter 18 (https://pages.cs.wisc.edu/~remzi/OSTEP/vm-paging.pdf)
7	10/1	Page Tables and TLBs	OSTEP: Chapter 19 (https://pages.cs.wisc.edu/~remzi/OSTEP/vm-tlbs.pdf)
	10/3	Demand Paging	• OSTEP: Chapter 21 (http://pages.cs.wisc.edu/~remzi/OSTEP/vm-beyondphys.pdf) • (http://pages.cs.wisc.edu/~remzi/OSTEP/vm-beyondphys.pdf) OSTEP: Chapter 22 (http://pages.cs.wisc.edu/~remzi/OSTEP/vm-beyondphys-policy.pdf)
8	10/8	Memory allocation and free space management	OSTEP: Chapter 17 (http://pages.cs.wisc.edu/~remzi/OSTEP/vm-freespace.pdf)
	10/10	COW and other cool tricks	OSTEP: Chapter 23 (https://pages.cs.wisc.edu/~remzi/OSTEP/vm-complete.pdf)
		Part III: Concurrency	
9	10/15	Intro to threads and concurrency	OSTEP: Chapter 26 (http://pages.cs.wisc.edu/~remzi/OSTEP/threads-intro.pdf)
	10/17	Locks and lock-based concurrent data structures	• OSTEP: Chapter 28 (http://pages.cs.wisc.edu/~remzi/OSTEP/threads-locks.pdf) • (http://pages.cs.wisc.edu/~remzi/OSTEP/threads-locks.pdf) OSTEP: Chapter 29 (https://pages.cs.wisc.edu/~remzi/OSTEP/threads-locks-usage.pdf)
10	10/22	Condition variables	OSTEP: Chapter 30 (http://pages.cs.wisc.edu/~remzi/OSTEP/threads-cv.pdf)
	10/24	Semaphores	OSTEP: Chapter 31 (http://pages.cs.wisc.edu/~remzi/OSTEP/threads-sema.pdf)
11	10/29	Concurrency bugs	OSTEP: Chapter 32 (http://pages.cs.wisc.edu/~remzi/OSTEP/threads-bugs.pdf)
	10/31	**Midterm II**	
		Part IV: I/O and Filesystems	
12	11/5	Files and directories	OSTEP: Chapter 39 (http://pages.cs.wisc.edu/~remzi/OSTEP/file-intro.pdf)
	11/7	Hard disks and RAID	• OSTEP: Chapter 37 (http://pages.cs.wisc.edu/~remzi/OSTEP/file-disks.pdf) • (http://pages.cs.wisc.edu/~remzi/OSTEP/file-disks.pdf) OSTEP: Chapter 38 (https://pages.cs.wisc.edu/~remzi/OSTEP/file-raid.pdf)
13	11/12	Filesystem Implementation	OSTEP: Chapter 40 (https://pages.cs.wisc.edu/~remzi/OSTEP/file-implementation.pdf)
	11/14	Fast filesystem (FFS)	OSTEP: Chapter 41 (https://pages.cs.wisc.edu/~remzi/OSTEP/file-ffs.pdf)
14	11/19	Crash consistency (FSCK)	OSTEP: Chapter 42 (https://pages.cs.wisc.edu/~remzi/OSTEP/file-journaling.pdf)
	11/21	Log structured filesystem (LFS)	OSTEP: Chapter 43 (https://pages.cs.wisc.edu/~remzi/OSTEP/file-lfs.pdf)

15	11/26	***Special Topic (TBD)***	
	11/28	***Thanksgiving Holiday (No Class)***	
16	12/3	Flash-based SSDs	**OSTEP: Chapter 44** **(https://pages.cs.wisc.edu/~remzi/OSTEP/file-ssd.pdf)**
	12/5	Review, Wrap-up	
	12/13	***Final Exam***: 9:45a-12p	