CS4328 & CS5305: Operating Systems

Spring 2026

Instructor:
Mina Guirguis
Email: msg@txstate.edu
Phone: (512) 245-6384
Office: Comal building (office #307.D.1)

Instructor Office Hours:
Mondays: 8:30 am – 10:30 am (Comal #307.D.1)
Wednesdays: 3:30 pm – 5:00 pm (Comal #307.D.1)

Teaching Assistant:
Md Khairul Azman
Email: khairul.azman@txstate.edu

Teaching Assistant Office hours:
Tuesdays: 4:00 pm – 6:00 pm (over zoom)
Thursdays: 4:00 pm – 6:00 pm (over zoom)

Zoom link:

https://txstate.zoom.us/j/87252978277

Class Material:
Maintained on Canvas. Homework, projects, and other documents will be uploaded under Files. An email will be sent out anytime a file (assignment, project, class note) is added.

Copyrights:
All class material provided in-class and on the Canvas course site are protected by Copyright and for the exclusive use of students enrolled in this course. Allowing others to access this material by placing it on publicly available git repositories or submitting to "note sharing sites" such as Chegg and CourseHero (which encourage you to break the law and post copyrighted content you don't own) is expressly forbidden. Note that you are not allowed to publicly post any of this material even if you made modifications to it. This copyright protection extends past the end of the semester.

Lectures:
Mondays: 2:00 pm - 3:20 pm (DERR #234)
Wednesdays: 2:00 pm – 3:20 pm (DERR #234)

This class meets face-to-face.

Textbook:
Operating Systems Concepts, Silberschatz, Galvin and Gagne, 10th Edition.

Grading Policy:
Midterm exam (30%)
Final exam (40%)
Written assignments ~ 4 (10%)

Programming assignments ~ 2 (20%)

Written assignments are due by midnight on the assignment due date and will be submitted through Canvas. Late written assignments will not be accepted without prior arrangements. It is your responsibility to ensure that you have uploaded the correct assignment. Programming assignments are also due by midnight on the assignment due date and are to be submitted through Canvas. Late programming assignments would incur a penalty of 10% per day for up to 2 days, and then they will not be accepted. No makeup exams and/or extensions are allowed without prior arrangement.

The Teaching Assistant (TA) oversees the written assignments and projects and their grading aspects. Please reach out to them with any grading issues first and if an issue is not resolved, please reach out to the instructor.

Usage of AI and generative AI:
For this class, "AI" has a broad definition and refers to any application or website that creates content including creating code. The use of most AI or Generative AI tools, including those provided by Texas State University, is limited in this course. Using AI to solve homework problems and write code for project assignments is not allowed and will be treated as any violation of the Academic Honor Code as explained below.

You may use AI to check for certain programming syntax, understanding the nature of errors that can occur while compiling and running programs. It is your responsibility to clearly understand any assistance received from AI that you adopt.

Academic Honesty:
Discussion of course material, problem sets, and projects are encouraged between students. However, you must write up your answers on your own. You must also write the names of other students you discussed any problem with. As for programming assignments, you may still discuss them with other students, however, what you submit should be strictly the code that you wrote. Submitting someone's else code/work (including AI-generated ones) will result in a final grade of 'F' and a referral to the Chair of the Texas State University Honor Code Council (HCC).

Midterm and Final exam Policy: The use of smart phones during exams is strictly prohibited and phones must be turned off and stored away. Any phone that is turned on during an exam will result in a grade of zero in the exam.

Please refer to the Code of student conduct:

https://studenthandbook.txst.edu/rules-and-policies.html

and the Honor Code:

https://www.txst.edu/honorcodecouncil/Academic-Integrity.html

Mission:
Texas State University is a public, student-centered research institution dedicated to excellence, discovery, and innovation. We create new knowledge, embrace a diversity of people and ideas, foster cultural and economic development, and educate our students to participate fully and freely in the communities of Texas, the nation, and the world.

Shared Values:

In pursuing our mission, we, the faculty, staff, and students of Texas State University, are guided by a shared collection of values:

- Teaching and learning based on research, student needs, and the free exchange of ideas in a supportive environment;
- Research and creative activities that encompass the full range of academic disciplines;
- Meaningful student engagement built on active involvement, accessibility, and intentional educational experiences;
- The cultivation of university community that consistently practices integrity, civility, compassion, and respect;
- A shared commitment to creating a sense of belonging across unique communities, identities, ideas, and contributions;
- A welcoming spirit and a global perspective;
- Dedication to service and leadership for the public good;
- Responsible stewardship of our resources and environment; and
- Continued reflection and evaluation to ensure that our strengths always benefit those we serve locally and globally.

Course Description:
This course is about the design of operating systems.

Course Content:
- Computer system overview.
- Process description and control.
- Concurrency issues (mutual exclusion and synchronization).
- Deadlock prevention, avoidance and detection.
- Memory management and paging.
- Virtual memory.
- CPU scheduling algorithms.
- Disk scheduling algorithms.
- File management.
- Virtual machines.
- Other topics (security, embedded systems, distributed systems and cloud computing).

Learning Objectives:
- Explain the objectives and functions of modern operating systems.
- Know how operating systems have evolved over time from primitive batch systems to sophisticated multi-user systems.
- Discuss the advantages and disadvantages of using interrupt processing.
- Describe the need for concurrency.
- Demonstrate the potential run-time problems arising from the concurrent operation of many separate tasks.
- Create state and transition diagrams for simple problem domains.
- Explain conditions that lead to deadlocks.
- Compare and contrast the common algorithms used for both preemptive and non-preemptive scheduling of tasks in operating systems.
- Describe relationships between scheduling algorithms and application domains.
- Discuss the need for preemption and deadline scheduling.
- Explain memory hierarchy and cost-performance tradeoffs.

- Summarize the principles of virtual memory as applied to caching, paging, and segmentation.
- Describe the performance metrics used to determine how a system performs.
- Compare and contrast the common algorithms used for disk scheduling.
- Analyze the performance of different implementations in a variety of settings via simulation experiments.
- Implement system calls.

Prerequisites:
CS 2318 Assembly Language
CS 3358 Data Structures

Accommodations:
If you require any special accommodation(s), please make sure to let me know during the first two weeks of the semester.

Roadmap:

This is a tentative roadmap and is subject to change!

Date	Topic(s)	Readings	Assignments (date assigned)
1/21/2026	Syllabus. Computer hardware and interrupts.	1 - 1.2	
1/26/2026	Computer System architectures.	1.3 - 1.6	
1/28/2026	System calls. Processes. Process State.	2.1, 2.3, 2.5 2.8-2.8.5	
2/2/2026	Process state.	3 - 3.3	
2/4/2026	Interprocess communication. Definitions.	3.4-3.6	Homework 1
1/9/2026	Multithreading.	4 – 4.3, 4.6	
2/11/2026	Scheduling Algorithms.	5 - 5.3	
2/16/2026	Scheduling Algorithms.		Homework 2
2/18/2026	Real-Time Scheduling.	5.6	
2/23/2026	Discrete-time Event Simulation.	Class notes	Project 1
2/25/2026	Process Synchronization.	6 - 6.6	
3/2/2026	Process Synchronization.		
3/4/2026	Classic Synchronization Problems.	7 – 7.1	
3/9/2026	Review		
3/11/2026	Midterm exam		
3/23/2026	Classic Synchronization Problems.		
3/25/2026	Deadlocks.	8 - 8.2	
3/30/2026	Deadlocks.	8.3 - 8.5	Homework 3
4/1/2026	Deadlocks.	8.6 - 8.8	Project 2
4/6/2026	Memory Management.	9 - 9.4	
4/8/2026	Memory Management.	9.5	
4/13/2026	Virtual Memory.	10 – 10.5	
4/15/2026	Virtual Memory.	10.5 - 10.9	
4/20/2026	I/O and Disk Scheduling.	11 - 11.4	Homework 4
4/22/2026	I/O and Disk Scheduling.		
4/27/2026	Networks and distributed systems.	19 -19.5	
4/29/2026	Virtual Machines and Containers.	18 – 18.5	
5/4/2026	Cloud-native OS.	Class notes	
5/6/2026	Review.		
5/11/2026	Final exam (2pm – 4pm)		