Syllabus, as of Feb 5, 2025

CS 4375: Operating Systems Concepts

Spring 2025

Instructor:	Nigel Ward, nigel@utep.edu, CCSB 3.0408 or 915-747-6827, not in Teams Office Hours: Mondays 3:00 – 4:00, Wednesdays 2:30 – 3:30, and by appointment
TA:	Chris Lawson, calawson@miners.utep.edu , Office Hours:Tu 9-11 and Th 9-11 and 2:30-3:30, and by appointment, in CCSB 1.0706
Class Time:	MoWe 9:00 – 10:20, Psych 308
Textbooks:	*Operating Systems: Three Easy Pieces*, by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau. *Computer Networks: A Systems Approach, 6th Ed*, by Larry Peterson and Bruce Davie.

Course Description: Process and thread management, processor scheduling and concurrency, interprocess communication, memory management, input/output management, file systems, and networking basics

Goal: Introduce concepts that will be foundational for further study, whether academic or professional, of 1) Computer Security and Forensics, 2) Systems Administration and Network Administration, and 3) Systems Programming and Network Programming, including developing for embedded systems, cloud systems, and high-performance systems.

Course Policies

The prerequisite for this class is CS 3432 (Computer Organization) with a C or better. This may be enforced retroactively.

Assigned readings are to be done before class.

Bring to class every day the relevant textbook, either hardcopy or on your device, or on your buddy's device.

Assignments are to be submitted in Blackboard or inhardcopy, as specified. The maximum points attainable for late assignments will be reduced by 10% per day or partial day of lateness, for up to five days; or more if the lateness impedes participation in class or the assignment is received after the solution has been discussed. Email submissions of assignments are not accepted.

Assignments are to be done individually unless specifically designated as group assignments. While you may discuss assignments with others, and you may share testing data and strategies, your solutions should be designed, written/integrated, and tested by you alone. If you need help, consult the TA or the instructor.

The use of found code and shared code is acceptable, unless otherwise specified, provided that you acknowledge your sources, state specifically what you used, and understand every line of code.

Programming assignments will be graded primarily on functionality, design quality, thoroughness of testing, and readability, and for those with demos, ability to discuss. Some of these factors involve subjective judgments; if you have questions about the criteria, please see the TA or the instructor.

Tests will be closed-book, except that one single-sided page of hand-written notes may be used for the first test, two for the second test, and three for the final. If you leave the room for any reason, your test will be graded on only what you did up until that time. No make-up exams will be given except under the conditions set forth in the Catalog, https://catalog.utep.edu/policies-regulations/attendance-grading/ . The lowest quiz grade will be dropped.

Grades will be based on two components: tests and the final examination, and assignments and other factors. Balanced performance across both components is important, so the final grade will be based on the order-weighted average of the two, with the weakest score's weight being twice that of the strongest score. Within the first component, each test will count for 28% and the final for 44%. Within the second component, the assignments will count for about 90%, with the remainder based on quizzes, in-class exercises, participation, and, very rarely, extra credit, notably for being the first to notify us of a problem with an assignment, or for producing top-quality code that we find suitable for sharing with the class as an exemplar.

Assignments and tests will be challenging. Grading will be on a points-earned basis (points above zero), rather than a points-off basis (points below expectation). Letter grades will be assigned accordingly: the A/B break will be 80% or lower and the B/C break 70% or lower.

Students are free to attend class or not, bearing in mind that absence may annoy other students, interfere with learning, and result in a lower grade.

General Policies

No make-up exams or assignments will be given except under the conditions set forth in the Catalog.

Students are expected to be punctual and to conduct themselves professionally and courteously.

If you have or suspect a disability and need accommodation, contact the Center for Accommodations and Support Services at 747-5148 or at cass@utep.edu or visit Room 106 Union East.

Topics, Readings and Major Assignments, tentative

Introduction (3 days)		ADAD Ch. 1-4
	Python warm-up asst 3%	
Processes (3 days)		ADAD Ch 5-7, 11
	Shell assignments 20%	
Memory (3 days)		ADAD Ch 12-18
	sundry exercises 5%	
	Test 1	
Sockets (3 days)		various readings
	Socket assignments 15%	
Threads and Concurrency (5 days)		ADAD 25 - 26; portions of 28 - 32
	Concurrency assts 30%	
	Test 2	
Networking (5 days)		ADAD 48; PD Ch 1, §2.1, §3.1 thru 3.1.1, §3.3 thru 3.3.5, Ch 5 thru 5.2.3, §7.1, 8.5.6, §9.3.1; ADAD 56
	Networking assts 12%	
Files and Devices (4 days)		ADAD Ch. 35-37, 39-40; portions of 38-46
	File System assts 8%	

Important Dates

January 22: first day
February 24 (tentative): Test 1
March 10-14: Spring Break
April 7 (tentative): Test 2
May 7: Review Day
May 14: Final Exam, 10:00-12:45

Course Website: http://www.cs.utep.edu/nigel/os/

Level 1: Knowledge and Comprehension

Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. The material has been presented only at a superficial level.

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

1. Choose a scheduling approach suitable for a given simple problem.
2. Explain segmentation and its security implications.
3. Explain some ways in which virtualization creates vulnerabilities.
4. Explain the components of process and virtual machine context.
5. Explain the need for paging and the basics of demand loading.
6. Describe the motivation for and gross characteristics of a trusted computing base.
7. Explain how domain names, IP addresses, file names, and memory segments are handled.
8. Given an application, identify the factors relevant to choosing a synchronous or asynchronous solution.
9. Choose when to use datagram versus virtual-circuit communication.
10. Differentiate transmission and propagation latencies and some factors affecting them.
11. Explain how data is serialized (byte order, representation, buffering).
12. Interpret the output of a packet capture tool.
13. Explain the role of cryptographic hashes and symmetric and asymmetric keys in security.
14. Explain the basic concepts of the Domain Name System (DNS) and the Internet Protocol (IP)
15. Explain the functionality handled at different network layers.
16. Explain the memory hierarchy and the basic concepts of distributed storage.
17. Explain generic device APIs, including the bidirectional handling of interrupts and requests.

2.2 Level 2: Application and Analysis

Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details.

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

18. Use the concepts of process state and state transition to characterize system and process behavior.
19. Relate the distinction between supervisor and user permissions to the design and implementation of system calls.
20. Write programs that use interprocess communication, specifically pipes and/or sockets.
21. Use simple system calls for common needs.
22. Implement producer-consumer coordination.
23. Build a server-side program that uses multi-threading to handle multiple simultaneous clients.
24. Identify situations where deadlock may occur and suggest ways to prevent it.
25. Perform simple arithmetic computations related to major families (for example determine page number or whether an address is within a power-of-2 segment).
26. Correctly use semaphores or condition variables for simple problems.

2.3 Level 3: Synthesis and Evaluation

Level 3 outcomes are those in which the student can apply the material in new situations.
This is the highest level of mastery.

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

27. When a process or a computer is running too slowly, infer some probable causes.
28. Choose among virtual machines, processes, containers and sandboxes as ways to support common programmer needs.
29. Distinguish when blocking versus nonblocking calls are appropriate.

Student Comments on previous offerings of this course

Fall 2024

1. Dr Ward has been one of the best instructors I have had at UTEP so far. He is strict when it comes to the syllabus, however he makes sure to engage all of his students to participate in discussions and answer questions. He also uses time throughout the semester to see what his class thinks could help improve the course and he takes all the suggestions into consideration. Overall one of my favorite classes I have taken.

2. One project a week, not two to be able to focus on other classes.

3. In class activities helped immensely with learning experience. Very involved class, which results in effective communication between peers and professor. 10/10 would take again.

4. I am unsure if I will pass this course, but besides that this professor taught me a lot and really helped me build upon my foundation with software development. If I fail I would definitely try to sign up with him again.

5. I like how Dr. Ward has well-prepared classes; he is an excellent instructor.

6. Ward was very consistent with what we did: readings, assignments, quizzes. My only discerning comment is on his networking book. Being very new to networks, the book used is very intimidating and too in-detail when there are only tidbits expected to be extracted. Overall, this course was amazing and the professor did an excellent job. Specifically I want to highlight Ward's ability to answer student's (mine included) very poorly worded questions.

7. The level of class participation maintained during lectures was spot on. It kept me engaged throughout the course.

8. He is a very good instructor, its just that it feels like a brutal climb to maintain a good grade in the class sometimes.

9. It was a really hard class for me in general, I hope he gives us extra points/partial credit on the final exam or something. We (I) really need it. Aside from that, he is a great professor, really interactive and passionate about what he teaches.

10. Ward is cool!!

11. Professor Ward taught me a lot and I highly recommend taking operating systems concepts with him. Excellent class

12. N/A

13. Professor is awesome, and really made the class really enjoyable.

14. Dr. Ward made the class interesting, easy to follow along in.

Spring 2023

1. Dr. Ward is a great instructor. Very straight forward and easy to understand.

2. my favorite class this semster. Lecture struture is different and not boring like other classes.

3. The professor is great at explaining complicated subjects. However, the assignment instructions could be clearer. Thankfully, the professor was quick to answer questions and very helpful. This was definitely one of the best CS classes I have taken at UTEP

4. Dr.Ward was consistent with daily quizzes, daily readings, and followed the syllabus in a consistent manner when it came to the assignments. Some assignments could've been more detailed and better described in the instructions, some assignments weren't clear. …

5. I really appreciated the clear structure of the course, which made it easy for me to prepare for each class.