Syllabus, as of January 2, 2023

CS 4375: Theory of Operating Systems

Spring 2023

Instructor:	Nigel Ward, nigel@utep.edu Office Hours: TBD, or by appointment, Computer Science 3.0408, also 747-6827
TA:	TBD
Class Time:	MW 1:30 – 2:50, CCSB 1.0202
Textbooks:	*Operating Systems: Three Easy Pieces*, by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau. *Computer Networks: A Systems Approach,6th Edition,* 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 with a C or better.

Assigned readings are to be done before class.

Bring to class a whiteboard marker or two and the relevant textbook, either hardcopy or on your device, or on your buddy's device.

Assignments are to be submitted in hardcopy, unless otherwise 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 unless otherwise specified.

Assignments are to be done individually unless specifically designated as group assignments. While you may discuss assignments with others, 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 functionally, design quality, thoroughness of testing, and readability. Some of these factors inevitably 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, assignments quizzes, etc. will be given except under the conditions set forth in the Catalog. The lowest quiz grade will be dropped.

Grades will be based on four components, weighted approximately as follows: 45% assignments, 22% final examination, 28% tests, and 5% other factors, including quizzes, in-class exercises, and participation.

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 probably be around 80% and the B/C break around 70%. Final grades may be adjusted upwards in cases where performance on both the assignments and the tests is solid.

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, as always, 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 assignment	
Processes (3 days)		ADAD Ch 5-7, 11
	Shell assignment	
Memory (3 days)		ADAD Ch 12-18
	sundry exercises	
	Test 1	
Sockets (3 days)		various readings
	Sockets assignments	
Threads and Concurrency (5 days)		ADAD 25 - 26; 28 - 32, portions
	various assignments	
	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, Ch 8 thru 8.2, §8.5.6, §9.3.1
	sundry exercises	
Files and Devices (4 days)		ADAD Ch. 35-37, 39-40; 38-46 portions
	sundry exercises	
	Final Exam	

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

Important Dates (tentative)

January 18: Class begins
February 15: Test 1
March 27: Test 2
May 5: Dead Day
May 10: Final Exam, 6:00-6:45

Level 1 Outcomes: 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:

V1i. Choose a scheduling approach suitable for given simple problem.
V1j. Explain segmentation and its security implications.
V1l. Explain some ways in which virtualization creates vulnerabilities.
V1m. Describe and motivate the components of process and virtual machine context..
V1n. Explain the need for paging and the basics of demand loading.
V1o. Describe the motivation for and gross characteristics of a trusted computing base.
V1x. Explain how domain names, IP addresses, and file names work and why.
C1c. Given an application, identify the factors relevant to choosing a synchronous or asynchronous solution.
E1f. Choose when to use datagram versus virtual-circuit communication.
E1h. Differentiate transmission and propagation latencies and some factors affecting them.
E1i. Explain how data is serialized (byte order, representation, buffering).
E1j. Explain the difference between lossy and lossless compression.
E1l. Interpret the output of a packet capture tool.
E1n. Explain the role of cryptographic hashes and symmetric and assymetric keys in security.
E1o. Explain the functionality handled at the physical, link, network, and transport layers.
E1p. Explain the functionality handled at different network layers.
E1q. Explain some data structures used for storing files on disk.
E1r. Explain the memory hierarchy, locality, and redundancy.
E1s. Explain generic device APIs, including the bidirectional handling of interrupts and requests.

Level 2 Outcomes: 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:

V2q. Use the concepts of process state and state transition to characterize system and process behavior.
V2r. Relate the distinction between supervisor and user permissions to the design and implementation of system calls.
V2t. Write programs that use interprocess communication, specifically pipes and/or sockets.
V2u. Use simple system calls for common needs.
C2g. Implement producer-consumer coordination.
C2h. Build a server-side program that uses multi-threading to handle multiple simultaneous clients.
C2i. Identify situations where deadlock may occur, and suggest ways to prevent it.
A2g. Perform simple arithmetic computations related to major families (e.g. determine page number or whether an address is within a power-of-2 segment)

Level 3 Outcomes: 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:

V3p. Choose among virtual machines, processes, containers and sandboxes as ways to support common programmer needs.
V3w. When a process or a computer is running too slowly, infer some probable causes.
C3j. Distinguish when blocking vs nonblocking calls are appropriate.