CS 4375: Operating Systems Concepts
Spring 2024
Syllabus

Title:	CS 4375: Operating Systems Concepts, CRN 23915
Term:	Spring 2024
Duration:	Tue 01/16/2024 through Thu 05/02/2024
Lecture Time:	Tue, Thu, 09:00AM - 10:20AM
Lecture Location:	Education Building 112
Instructor:	Dr. Christoph Lauter `cqlauter@utep.edu` 915-747-5939
TA:	see class website
Instructors Office Hours and Location:	Email anytime, Phone during business hours, Office Mon-Thur 10:30AM-11:45AM
Class Web page:	`https://www.christoph-lauter.org/utep-os/`
Prerequisites:	CS 3432 with a grade of C or better
Description:	CS 4375 is a course about the systems-level software called the operating system that provides an interface between application software and the computer hardware. The operation system is responsible for sharing resources, providing common services, and protecting programs from interference from other programs. Topics covered in the course include process and thread management, processor scheduling and concurrency, interprocess communication, memory management, input/output management, file systems, and networking basics.
Student Learning Outcomes:	See appendix below. Student progress on the learning outcomes will be assessed primarily via programming projects (homework) and exams.
Textbooks:	Required: • Tanenbaum, Andrew S., and Herbert Bos. Modern operating systems. Prentice Hall Press, 2014. (4th edition) Optional: • Stallings, William. Operating Systems: Internals and Design Principles, Edition: 8. Pearson, 2014. • Silberschatz, Abraham, Peter Baer Galvin, and Greg Gagne. "Operating System Concepts", John Wiley and Sons Inc., 2012. • Kerrisk, Michael: The Linux Programming Interface, `https://nostarch.com/tlpi`.

Homework Assignments: There will be 3 homework assignments. The assignments are team assignments, to be solved by teams of two students. The assignment require programming in C (not C++) and producing portable, quality code with all peripheral software products required, such as makefiles. Homework submitted without a write-up in the form of a PDF file will be downgraded by 50%. The write-up must be in English and must be submitted as a PDF file (no Microsoft Word). Code submitted for Homework which does not compile (with `gcc`) will not be considered for grading by the instructor. Late homework will be downgraded 30% per day late.

What is called "homework" by your instructor might be called "project" by other instructors. **The homework assignments given in this course are long and non-trivial. Start working with your team right away. Progress reports must be given in class every week by every team.**

Workload & Attendance: This course requires participation during the in-class lectures, in addition to the daily readings and the three homeworks. Students must read the textbook before the corresponding class. The instructor rarely uses slides but rather writes important information on the whiteboard; students must take notes in class for reference in exams. Attendance in the lecture will be checked. This course is programming-intensive and requires student committment to understanding low-level concepts and complex data-structures. Students are encouraged to favor simple and working solutions over complex and incomplete approaches.

Please form study groups and you are encouraged to discuss the approach and understand the problem. The write up, programming, and actual solutions to the homeworks are team-work but must be the individual work of the team. If you use someone's work for your own, you are committing plagiarism.

Readings: Students are expected to read and understand the textbook chapters before these chapters are discussed in the lecture.

Computers: Students need access to a computer that allows them using a native Linux environment or that allows them `ssh` access to a Linux server that will be provided by the instructor. For homework assignments that compile in one environment and not the other, the result obtained on the system provided by the instructor will be deemed the canonical result.

Tests and Testing Policy: Three tests will be given: two mid-term tests and a final. All tests are cumulative, with an emphasis on recent material. No make up tests are given but for documented medical emergencies.

Grading:

Letter grade:

Points range	Grade
$[90; 100]$	A
$[80; 90)$	B
$[70; 80)$	C
$[60; 70)$	D
$[0; 60)$	F

Grading breakdown:

Homework	60%
Midterm I	7%
Midterm II	8%
Final exam	20%
Attendance	5%

Make-Ups:

There will be no make-up for missed exams or homework deadlines, unless due to a documented medical or family emergency. Providing the right kind of documentation for emergencies is the student's responsibility.

Approximate Course Schedule:

The course will be spread out **approximatively** according the following schedule:

Class 1	Introduction: What's an OS, OS types
Class 2	C programming
Class 3	C programming, system calls
Class 4	C programming, assembly, interruptions
Class 5	History of OSes, Hardware overview
Class 6	Modern OSes
Class 7	Processes, threads
Class 8	IPC, semaphores, mutexes
Class 9	Monitors, RPC, locks, scheduling
Class 10	Review
Class 11	Midterm I
Class 12	Memory, VMs
Class 13	Paging, implementation, segmentation
Class 14	File System overview
Class 15	File System implementation
Class 16	(cont'd)
Class 17	I/O, Memory Mapped, Direct Access
Class 18	(cont'd)
Class 19	Review
Class 20	Midterm II

Class 21	Deadlocks
Class 22	(cont'd)
Class 23	Lightweight Processes, Thread Programming
Class 24	(cont'd)
Class 25	Intro to Networking, Socket Programming
Class 26	(cont'd)
Class 27	Review

This course schedule is **approximative**. In the case the instructor thinks that students need more time to better understand a subject (or less time because they are really, really smart), the instructor will deviate from the schedule. This may affect the dates the midterm exams are given.

Academic Honesty:

- Students are expected to conduct themselves in a professional and courteous manner, as prescribed by the Standards of Conduct: `https://www.utep.edu/hoop/section-2/studentconduct-and-discipline.html`
- Submitted work should be unmistakably your own. You may not transcribe or copy a solution taken from another person, book, or other source (e.g., a web page). Professors are required to report academic dishonesty and any other violation of the Standards of Conduct to the Dean of Students.
- Permitted collaboration: Students may discuss requirements, background information, test sets, solution strategies, and the output of their programs. However, implementations and documentation must be their own creative work. Students are required to document advice received from others and all resources utilized in the preparation of their assignments.
- If academic dishonesty is suspected: The Dean of Students office will be contacted for adjudication. A temporary "incomplete" grade will be issued if their investigation extends beyond the grading period.

Disabilities & Accomodations: If you have a disability and need accommodations, please contact The Center for Accommodations and Support Services (CASS) at 915-747-5148, or by email to `cass@utep.edu`. For additional information, please visit the CASS website at `www.sa.utep.edu/cass`.

COVID19 statement:

- Please stay home if you have been diagnosed with COVID-19 or are experiencing COVID-19 symptoms. If you are feeling unwell, please let me know as soon as possible, so that we can work on appropriate accommodations. If you have tested positive for COVID-19, you are encouraged to report your results to `covidaction@utep.edu`, so that the Dean of Students Office can provide you with support and help with communication with your professors.
- The Center for Disease Control and Prevention recommends that people in areas of substantial or high COVID-19 transmission wear face masks when indoors in groups of people. The best way that Miners can take care of Miners is to get the vaccine. If you still need the vaccine, it is widely available in the El Paso area. For more information about the current infection rates, testing, and vaccinations, please visit `epstrong.org`.

Cell phones: Please silence your mobile devices or put them into a vibrate mode for the duration of class - they are disruptive for your fellow students. No mobile devices (cell phones, PDAs, laptops etc.) are allowed during the exams and will result in your expulsion from the test.

Appendix: CS 4375 Learning Outcomes

Level 1:

- 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. Explain 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, file names, and memory segments are handled.
- 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).
- E1l. Interpret the output of a packet capture tool.
- E1n. Explain the role of cryptographic hashes and symmetric and asymmetric keys in security.
- E1o. Explain domains names, IP and MAC addressing and how they support administration and data locality.
- E1p. Explain the functionality handled at different network layers.
- E1q. Explain some concepts in storing files on disk.
- E1r. Explain the memory hierarchy and the basic concepts of distributed storage.
- E1s. Explain generic device APIs, including the bidirectional handling of interrupts and requests.

Level 2:

- 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 (for example, determine page number or whether an address is within a power-of-2 segment)

Level 3:

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