CS444: An Introduction to Operating Systems
FALL 2025

Dr. J. Holly DeBlois
Office: McCormack, 3rd floor, room M-3-201-32
Office hours: Tues/Wed/Thurs 12:30-1:30pm

Lectures	Tuesday & Thursday, 2:00-3:15, W01-0004 section 01 (course 1431) Tuesday & Thursday, 5:30-6:45, M02-0404 section 02 (course 4046)
Instructor Email	jane.deblois@umb.edu
Instructor Website	Lecture notes and assignments are at: https://www.cs.umb.edu/~hdeblois/ Grades are on canvas https://www.umb.edu/canvas/
Portal:	Register for cs444 at https://portal.cs.umb.edu/ to create your course directory
Piazza:	Join Piazza: https://piazza.com/umb/fall2025/cs444

Course Description: In CS444, we present the basic aspects of operating systems, a layer of software between hardware and user software, and we code projects in C.

Textbook: The main reference in this course is *Modern Operating Systems, 5th edition.*, (Pearson, 2023) by Andrew S. Tanenbaum and Herbert Bos.

Attendance: Mandatory. **If you miss more than 5 classes, your final grade will be lowered: miss 6 or 7, lower by 1 step; miss 8 or more, lower by 2 steps.** See chart of grade steps on last page.

Topics: We shall cover the following topics:

- **Operating System** definitions and features
- **Processes** and scheduling algorithms, threads, process/thread synchronization, the critical region problem and semaphores
- **Memory management**, including cache memory, virtual memory, paging and segmentation
- **File systems**: implementation, management and optimization
- **Input/Output control**: buffering, spooling, disk management and disk access scheduling
- **Deadlocks**: detection, recovery, avoidance and prevention
- **Virtualization** and cloud computing
- **Multiple processor systems**
- **Security** threats and defenses
- **Linux/Android/Windows**: differences and similarities

I**n-Class Exercises:** You need to bring a computer to class and have a setup in place to access the CS Linux servers. We will handwrite C code. We will use emacs. We will compile and run on the server during class. These exercises will not be graded, but they count toward attendance and help with projects.

Note: No courses required by the CS major, minor or certificate may be taken pass/fail.

Prerequisites: CS310 Algorithms and CS341 Computer Architecture. You need to understand how to describe and implement algorithms. You need to understand architecture down to driver and hardware level. You will also need the material from CS240 Programming in C and from probability distributions in CS220 Applied Discrete Math. We offer optional ungraded hw0/0.5 to assure that you are in a good position to start this course.

Evaluation: Your grade in the course is determined by: four C code programming projects (40%), four homework assignments (40%), two tests (20%) and attendance (explained above). Submit your programming and homeworks by uploading them to the server and placing them in your course directory. Do not change the permissions on your course directory.

Programming projects: Design carefully. Upload, edit, compile and run on the server since C libraries differ. Follow the rules in the Academic integrity section below – give sources for and limit the amount of non-original C code you copy, copy and modify or generate. **Provide a minimum of three incremental submissions of each project on the server starting at least two days before the assignment is due.** This shows your development process. We test your code using Measure of Software Similarity (MOSS) provided by Stanford University. **Software similarity to other students' code greater than 45% earns 0 points.** We may call you in to explain your code.

Homeworks: Upload your homework to the server. Drawings in the homework are handwritten, photographed, and included in your typeset document converted to PDF.

Tests: The two tests are oral exams, answering the instructor's questions in a 30- or 20-minute session. For test1, students form teams of two. For test2, each student will have an individual session.

Collaboration policy: You are encouraged to ask questions in class and may consult with your fellow students about assignments. On C code assignments, do not stay "stuck" – ask for help after 10 minutes of trying to make progress – use piazza. Be careful not to help another student by giving code – keep your laptop closed while conversing.

Late Submission or Late Arrival: Homework and C code project submissions and tests are scheduled for precise times. **Late code submission loses 1% of the score per hour late.** Code is picked up by script so you must name it and position it correctly. Late test arrival means you will be given less of the test and miss those points.

Academic integrity will be strongly enforced. Your code and homework must be your own product, may include some code copied or modified from outside sources or generated by chatGPT, but must be sufficiently original to pass the MOSS criteria given in Evaluation. See https://www.umb.edu/academics/provost/academic-integrity. Since UMB does not have any guidance for putting sources in code yet, we will use the MIT guidance. See https://integrity.mit.edu/handbook/writing-code/ Document any chatGPT code you generated and used by citing the prompt you entered. We recommend you take steps to not allow chatGPT to store your results. You must limit your amount of code from an outside source and from chatGPT to meet the MOSS requirement stated above in Programming Projects.

Additional references:

The C Programming Language, 2nd edition (Prentice Hall, 1978) by Brian W. Kernighan and Dennis M. Ritchie, which presents C in 8 chapters.

Operating Systems Design and Implementation, 3rd edition (Pearson, 2006) by Andrew S.Tanenbaum and Albert S. Woodhull, which includes 392 pages of C code for the MINIX OS.

https://www.gnu.org/software/emacs/refcards/pdf/refcard.pdf which explains the emacs editor.

Accommodation: Section 504 of the Rehabilitation Act of 1973 offers guidelines and support for curriculum modifications and adaptations for students with documented disabilities. Contact the Ross Center at 617-287-7430 and please discuss your accommodations with the instructor.

Student Conduct: You must be honest in all conduct. The University presupposes that work for academic credit is the student's own and complies with University policies above and here: https://www.umb.edu/camp-life/dean-of-students/student-conduct-process/.

Syllabus and Schedule Subject to Change: The instructor reserves the right to change the syllabus when necessary and will let you know. Here is the tentative Schedule (28 classes over 15 weeks):

week	class/date	chapter	topic	assignment
1	#1-Tue 9/2	syllabus	Introduction	hw 0/0.5 ungraded
		1	Overview of OS	In-class C code
	#2-Thu 9/4	1	OS, Huffman code	proj1, hwk1 posted
2	#3-Tue 9/9, add/drop ends	2	processes, threads, cmdline	In-class C code
	#4-Thurs 9/11	2	processes, threads, malloc	hw1 due
3	#5-Tue 9/16	3	memory, Hamming code	In-class C code
	#6-Thu 9/18	3	memory	In-class C code
4	#7-Tue 9/23	4	file systems	proj1 due proj2,hwk2 posted
	#8-Thu 9/25	4	file systems	In-class C code
5	#9-Tue 9/30	5	input/output	hwk2 due
	#10-Thu 10/2, NA grades due	5	input/output	In-class C code...
6	#11-Tue 10/7	12	designing an OS	
	#12-Thu 10/9	12	designing an OS/review	project2 due
7	#13-Tue 10/14	1-5,12	review(Ch 1-5, 12, C code)	
	Wed-Fri 10/15-17	testing	test 1 (Ch 1-5, 12, C code)	no class Thurs
8	#15-Tue 10/21	6	deadlock	proj3,hwk3 posted
	#16-Thu 10/23	6	deadlock	
9	#17-Tue 10/28	7	virtualization	hwk3 due
	#18-Thu 10/30	7	virtualization	
10	#19-Tue 11/4	8	multi-processor systems	
	#20-Thu 11/6	8	queuing theory,posix threads	project3 due
11	Tue 11/11, VeteransDay	no class	Holiday	
	#21-Thu 11/13	9	security	proj4,hwk4 posted
12	#22-Tue 11/18	9	security	
	#23-Thu 11/20	10	Unix, Linux, Android, Pass/Fail/WithdrawDeadline	hwk4 due
13	#24-Tue 11/25	10	Unix, Linux, Android	
	Thu-Sun 11/27-30	no class	Thanksgiving Recess	
14	#25-Tue 12/2	11	Windows	proj4due,testsignup
	#26-Thu 12/4	6-11	review(Ch 6-11, C code)	review
15	Mon -Wed 12/8-10	testing	test 2 (Ch 6-11, C code)	no class Tues
	#28-Thurs 12/11		special topic	course eval

Table of score to grade conversions (default grading scheme “UMB letter” in canvas):

93 <= S = A
90 <= S < 93 = A-
87 <= S < 90 = B+
83 <= S < 87 = B
80 <= S < 83 = B-
77 <= S < 80 = C+
73 <= S < 77 = C
70 <= S < 73 = C-
67 <= S < 70 = D+
63 <= S < 67 = D
60 <= S < 63 = D-
S < 60 = F