CS 444: Introduction to Operating Systems
SPRING 2025 apr21rev

Dr. J. Holly DeBlois
Office: McCormack, 3rd floor, room M-3-201-32
Office hours: Tues/Wed/Thurs 2:30-3:30pm

Lectures	Tuesday & Thursday, 4:00-5:15, W01-0088 section 02 Tuesday & Thursday, 5:30-6:45, W01-0088 section 01
Instructor Email	jane.deblois@umb.edu
Instructor Website	Lecture notes and assignments are at: https://www.cs.umb.edu/~hdeblois and grades are on canvas https://www.umb.edu/canvas/
Portal:	Register for cs444 at https://portal.cs.umb.edu to create your course directory

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, exploring machine-level and user-level applications. Most operating systems are written in C because the access to low-level functions is ideal. The main reference in this course is *Modern Operating Systems, 5th edition.*, (Pearson, 2023) by Andrew S Tanenbaum and Herbert Bos. 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

You need to bring a computer to class capable of accessing the CS Linux servers. We will write C code on the server, compile and run in class.

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

Prerequisites: CS 310 Algorithms and CS 341 Computer Architecture

Evaluation: The grade in the course is determined by: four programming projects, four homework assignments and two tests. Each of them is 10% of the score. Programming projects use the C language and the Linux servers of the department and must be edited and compiled there. Visual Studio (VS Code) as an editor is not ideal because you are not at the Command Line (CL) on the server, so some class time will cover emacs and possibly nano and vi. Homework must be typeset with software such as notepad, Microsoft Word and LATEX and converted to PDF for submission by uploading to the server (use secure copy command: see man scp). 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. Attendance is required, so missing more

than three classes entails points off, deducted from programming projects. In-class exercises will be collected and count as part of progrmming projects. The total score is converted to letter grades according to the table on the last page.

Academic integrity will be strongly enforced. Your code and homework must be your own product and may not be AI-generated. You may consult with your colleagues, but you must be the sole author of your submitted work. Your personal use of AI may need to be defined carefully to avoid conflicts, so please ask questions. See https://www.umb.edu/academics/provost/academic-integrity. Since UMB does not have any particular guidance for sourcing code yet, we will use the MIT guidance, which will be supplied and discussed in class, so we can be sure we know all the underlying versions of code that our code relies on.

Slides, projects, homeworks and scores will be posted on Blackboard and the Instructor's Webpage and work will be submitted on the CS server (in directories in your personal area /home and in class areas in /courses). Please read the assignments carefully when posted. Uploading your work to the server as you go along is required. You will need to use linux files, directories, groups and permissions. An ungraded hw0 will be provided to review your skills.

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/

Accommodation:

Section 504 of the Rehabilitation Act of 1973 offers guidelines and support for curriculum modifications and adaptations for students with documented disabilities. If applicable, students may obtain adaptation recommendations from the Ross Center for Disability Services, Campus Center, Upper Level, Room 0211, 617-287-7430. The student must present these recommendations and discuss them with each professor within a reasonable period, preferably by the end of Drop/Add period.

Student Conduct:

It is the expressed policy of the University that every aspect of academic life – not only formal coursework situations, but also all relationships and interactions connected to the educational process – shall be conducted in an absolutely and uncompromisingly honest manner. Homework 0 is designed to assure you, me and our graders that you are in a good position to start this course. The University presupposes that any submission of work for academic credit is the student's own and is in compliance with University policies, including its policies on appropriate citation and plagiarism. These policies are spelled out in the Code of Conduct https://www.umb.edu/campus-life/dean-of-students/student-conduct-process/.

Reserve Clause:

The instructor reserves the right to make changes in the syllabus when necessary to meet the learning objectives, to compensate for missed classes, schedule changes or hardware, software and network failures, or for similar legitimate reasons.

Tentative Schedule

week	date	chapter	topic	assignment
1	Tue 1/28	syllabus	Introduction	homework0
	Thu 1/30	1	Overview of OS	
2	Tue 2/4		Huffman code	proj1,hwk1 posted
	Thu 2/6		Snow Day	
	add/drop ends			
3	Tue 2/11	2	processes, threads	
	Thu 2/13	3	memory, malloc	hwk1 due
4	Tue 2/18		Hamming	
	Thu 2/20	3	memory	proj2,hwk2 posted
				2/21 proj1 due
5	Tue 2/25	4	file systems	
	Thu 2/27	4	file systems	hwk2 due
	NA grades due			
6	Tue 3/4	5	input/output	
	Thu 3/6	5	input/output	project2 due
7	Tue 3/11		review	
	Wed-fri 3/12-14		test 1 (Ch 1-5, 12)	
	Spring break			
8	Tue 3/25	6	deadlock	proj3,hwk3 posted
	Thu 3/27	6	deadlock	
9	Tue 4/1	7	virtualization	hwk3 due
	Thu 4/3	7	virtualization	
10	Tue 4/8	8	multi-processor systems	project3 due
	Thu 4/10	8	multi-processor systems	
11	Tue 4/15		queuing theory	
			POSIX threads	proj4,hwk4 posted
	Thu 4/17	9	security	
12	Tue 4/22	10	Unix, Linux, Android	proj4 revised
				hwk4 due
	Thu 4/24	10	Unix, Linux, Android	
13	Tue 4/29	11	Windows	proj4 due, post
				review, test signup
	Thu 5/1	11	Windows	
14	Tue 5/6	6-11	review	
	Wed 5/7		test 2 (Ch 6-11)	
	Thu 5/8		test 2 (Ch 6-11), no class	
	Fri 5/9		test 2 (Ch 6-11)	
15	Tues 5/13		last class	special topic
				course eval

90 <= S	A
87 <= S < 90	A-
84 <= S < 87	B+
81 <= S < 84	B
78 <= S < 81	B-
75 <= S < 78	C+
72 <= S < 75	C
69 <= S < 72	C-
66 <= S < 69	D+
63 <= S < 66	D
60 <= S < 63	D-
S < 60	F