CS 421: Operating Systems

Term: Spring 2022
Credits: 3
Meets: 9:40 AM – 11:10 AM, TTh in HS 309
Prerequisites: CS 212, CS 251, and CS 271, all with a grade of C or better.

Classroom Mask Requirement

All students are required to wear an appropriate mask that covers their mouth and nose when they're in the classroom. They must also adhere to additional expectations communicated by me or posted in the classroom. Note: UWO procedure dictates that, during the COVID-19 pandemic, I cannot begin class until all students are wearing a mask properly. If a student is non-compliant with the masking policy and refuses to leave the classroom promptly when requested, I am required to cancel class. Students responsible for class cancellation for these reasons will be referred to the Dean of Students office, and the student will be unable to attend class until they meet with the Dean of Students. The student may be dropped from the class by the Dean of Students.

Instructor

Instructor: Scott Summers
Email: `summerss@uwosh.edu`
Phone: 920-424-1324
Office: Halsey 220
Office hours: Indicated by the cells shaded in gray in the following table and subject to change.

	Mon	Tues	Wed	Thu	Fri
9:00 AM					
10:10					
11:10					
12:00 PM					
1:40					
2:40					

I encourage students to email me with your questions. Student should keep in mind that, when they email me, they should identify themselves and the class that you are taking with me before asking their question(s). It takes me a while to remember who is in what class. Emails should be written in a respectful and professional tone, using complete sentences. Note that I check my email quite frequently during the business week, but I don't tend to check it between about 7pm and 8am (I check it even less frequently on weekends).

Course Description (outdated)

An introduction to operating systems concepts. Topics covered include: interrupts, memory allocation, virtual memory techniques, process scheduling and synchronization, deadlocks, resource allocation, and file systems. A major programming project will be assigned to provide experience with operating system design.

Course Website

The course website is: `https://uwosh.edu/canvas/`. Students should check Canvas on a regular basis, perhaps two or three times per day.

Recommended Textbooks

- Operating Systems: Principles and Practice, Thomas Anderson and Michael Dahlin, 2nd edition, 2014, Recursive Books.
- Operating Systems: Three Easy Pieces, Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau. Available at `http://pages.cs.wisc.edu/~remzi/OSTEP/`

Course Grade

A student's final course grade will be based on the following components.

10% QUIZZES
There will be a quiz given roughly each week, each to be submitted online via Canvas. Each quiz must be completed individually and in isolation. Students shouldn't discuss quizzes with anyone other than the course instructor.

30% PROJECTS
There will be four programming projects given. Programming will be done using the C programming language in a Linux environment. Students may work with one other student in the class on each programming project.

60% EXAMS
There will be three, equally-weighted, in-class exams. Exam material will come from the lecture notes, quizzes and projects. There will be more information about each exam as it approaches. We will review for each exam during the lecture prior to the exam.

If a student is unable to take a scheduled exam, it may be possible to take a make-up exam provided that the student do both of the following, which are then subject to my approval:

1. Make arrangements prior to the scheduled exam (for last minute emergencies, telephone me at 920-424-1324 or leave a message at the Computer Science office, 920-424-2068. **No after-the-fact notifications will be accepted**.
2. Have a written medical excuse signed by the attending physician OR have a note of justification from the Dean of Students Office.

If allowed, only one make-up exam will be given. It will be a comprehensive exam given at an arranged time during the last week of the semester.

Late Work

Late work will NOT be accepted. Late work is worth 0 points. Extensions may be granted at the discretion of the instructor if the student provide's a valid justification (in the form of a written excuse from a medical doctor or the Dean of Students Office) before the due date.

Grading Scale

Grading will be on a plus/minus system. Grading may be done on a curve depending on the overall performance of the class. If no curve is used, then a student's grade will be computed based on the following:

Percentage	Grade
> 91	A
> 89 and ≤ 91	A-
> 87 and ≤ 89	B+
> 81 and ≤ 87	B
> 79 and ≤ 81	B-
> 77 and ≤ 79	C+
> 71 and ≤ 77	C
> 69 and ≤ 71	C-
> 67 and ≤ 69	D+
> 61 and ≤ 67	D
> 55 and ≤ 61	D-
≤ 55	F

Re-grading

If a student believes a quiz, project or exam was graded incorrectly or unfairly and would like to have it re-graded, then they must let me know about it in writing within one day of receiving the quiz, project or exam back. I will re-grade the entire quiz, project or exam and they may gain or lose points.

Course Objectives

1. Outcomes #1
   (a) Demonstrate by explaining or describing objectives and functions of modern operating systems
   (b) Demonstrate sound knowledge of the user and system views of the operating system and able to differentiate between these two viewpoints.
   (c) Adequately explain the structure and mode of operations of today's operating system as they relate to such concepts as multiprogramming, time-sharing, swapping, interrupt, and dual mode operation.
   (d) Demonstrate intimate knowledge of the issues of design and implementation as well as the desired user and system properties.
   (e) Show understanding of the influences of security, distributed and special-purpose systems, and different computing environments on modern operating system design.
   (f) Demonstrate the ability to discuss the tradeoffs inherent in operating system design and able to identify potential threats and their safeguards in designed systems.
2. Outcomes #2
   (a) Demonstrate ability to compare and contrast the many possible methods (e.g., simple, layered, modular, and microkernel) of structuring the operating system.
   (b) Show sound knowledge of the difference between asynchronous and synchronous interrupts and of the relative advantages of interrupts over polling.
   (c) Able to discuss the system call concept and able to differentiate between the needs for the system call interface and the application program interface.
   (d) Show how system programs manage the resources used by applications.

(e) Demonstrate the ability to illustratively explain why the services provided to users constitute a different set of functions than that provided to the system.

3. Outcomes #3

   (a) Able to explain process state and process control block using appropriate examples to highlight their components.

   (b) Able to illustratively describe the creation and termination, scheduling, and interprocess communication features of a process.

   (c) Show the ability to compare and contrast the various types of multithreading models.

   (d) Demonstrate a deep understanding of some of the issues encountered with multithreaded programs (e.g., `fork()` and `exec()` system calls, cancellation, signal handling, thread pools, thread specific data, and scheduler activations).

   (e) Able to describe possible run-time problems arising from the concurrent operation of many separate tasks.

   (f) Demonstrate the ability to explain how software and hardware are individually used to solve mutual exclusion problem.

   (g) Able to discuss the evaluation criteria for selecting a particular system's CPU scheduling algorithm and the various available criteria comparing CPU scheduling algorithms.

4. Outcomes #4

   (a) Able to compare and contrast paging and segmentation techniques.

   (b) Know how to evaluate the tradeoffs of the components parts in the memory hierarchy in terms of size, cost, and access time.

   (c) Able to compare and contrast demand paging and copy on write techniques.

   (d) Show the ability to analyze such techniques or policies as page replacement, swapping, and thrashing.

5. Outcomes #5

   (a) Able to discuss file system design tradeoffs including access methods, file sharing, file locking, directory structures, and protection.

   (b) Demonstrate sound knowledge and deep understanding of a file's attributes and operations.

   (c) Demonstrate a deep understanding of the details of a local file system and directory structure implementation.

University Policy on Academic Integrity

The University of Wisconsin Oshkosh is committed to a standard of academic integrity for all students. The system guidelines state:"Students are responsible for the honest completion and representation of their work, for the appropriate citation of source, and for respect of others' academic endeavors" (UWS 14.01, Wisconsin Administrative Code).

Statement Regarding Diversity, Equity & Inclusion

Diversity drives innovation, creativity, and progress. At the University of Wisconsin Oshkosh, the culture, identities, life experiences, unique abilities, and talents of every individual contribute to the foundation of our success. Creating and maintaining an inclusive and equitable environment is of paramount importance to us. This pursuit prepares all of us to be global citizens who will contribute to the betterment of the world. We are committed to a university culture that provides everyone with the opportunity to thrive.

Required Disclosure Statement

Students are advised to see the following URL for disclosures about essential consumer protection items required by the Students Right to Know Act of 1990:
`https://uwosh.edu/financialaid/consumer-information/`