

Cmp Sci 4760 – Operating Systems
Department of Mathematics and Computer Science
Fall 2025

Instructor Dr. Mark W. Hauschild
Classroom SSB 132
Office 327 ESH
Phone (314) 516-6426
e-mail hauschildm@umsl.edu

Classtime Mo/We 12:30pm – 1:45pm
Office Hrs Mo/We 1:45 – 2:15pm
Any other time by appointment
Prerequisites Cmp Sc 2750 and Cmp Sc 3130

Course Description This course covers the structure of a generic operating system, considering in detail the algorithms for interprocess communications, process scheduling, resource management, memory management, file systems, and device management. It presents examples from contemporary operating systems. This course also requires practical projects implemented within a modern operating system or simulator environment.

Required texts:

- Stallings. *Operating Systems: Internals and Design Principles*. Prentice-Hall, Englewood Cliffs, NJ. 2017.
- Robbins/Robbins. *Unix Systems Programming – Communication, Concurrency, and Threads*. Prentice-Hall, 2003.

Recommended text:

- Silberschatz. *Operating System Concepts*. Zybooks, 10th edition

Topics We'll discuss the structure of a generic operating system, with an emphasis on the algorithms and data structures needed to perform various tasks to manage processes, resources, memory, files, and IO devices. There will be *extensive* programming in C under Linux environment. The sequence of topics will be as follows (in that order):

- Function, Types, and Structure of Operating Systems.
- Concurrent Programming, Process Interaction and Synchronization.
- Process Management and Scheduling.
- Resource Management and Deadlock Problem.
- Memory Management.
- File System and I/O management.

Outcomes At the end of this course, you are expected to be able to proficiently perform system programming tasks. You should also be knowledgeable in different operating systems topics as outlined above. You should understand the algorithms to achieve various tasks in operating systems and use the knowledge to improve the performance of your programs.

Requirements This is an in-person course, so I expect to see you in class. Students who do not attend class almost always end up doing very poorly on the tests, as well as not doing well on the projects. It is especially important to keep up on projects and not fall behind. Missing classes or getting behind on projects to the point that you do not understand when I discuss the projects will definitely lead to a poor result.

Office hours will be either in-person or through zoom.

All correspondence should be made through your UMSL-provided email. Please use the class email list on Canvas for communications. Any unsigned email will go unanswered by me.

Projects: You will be given programming assignments, typically a set of programs every two weeks. Assignments will be due at 11:59pm on the day noted on the assignment. Assignments should be submitted on `hoare` and must execute properly on `opsys` for credit. You should start working on the projects as soon as they get assigned as some of them may get tricky. In the past, the students who have asked a lot of questions have scored better grades. Do not hesitate to ask a question in my office, or over email, especially if you do not have an idea on how to start working on the project.

Grading The grade will be based on programming assignments and three tests.

All tests are in-person and proctored. There are no circumstances under which you will be allowed to take an online exam in this course. If you have some accommodation or cannot make a test due to an emergency, you will be required to take a proctored test with myself or at the testing center.

Each assignment must be meticulously documented and clearly identify its purpose, author, and date. I will like to *read* your submitted code; I should *not have to figure it out*. It will do you good if you peruse the Gnu Coding Standards at <http://www.gnu.org/prep/standards/>. When you come to me for help with the code, or when you submit the code, make sure that you follow good indentation practices. If you miss any test or assignment without making prior arrangements, you will have a zero. I will not give any make up tests unless this is approved before the test.

You will be allowed to use generative AI for your project. However, this use must be documented and you are still required to understand how your code works and to be able to carefully describe how your project fulfills the assigned tasks. In addition to doing the required task, the projects must also meet output requirements.

I will also be doing code reviews with students. If you are given a code review, this review will determine your grade for the project. If you are unable to describe how your code works and it becomes obvious that you have not done your code or do not understand your own project, you will receive a 0. You will be of course doing the review with your code in front of you, so this is trivial if you did the project yourself.

The distribution of grades will be as follows:

Programming Assignments	40%
Three Tests	20% each

Failure to hand in any assignment will result in an automatic zero for that assignment. If some student is unable to hand in an assignment by the deadline, he/she must discuss it with me *before* the deadline. I'll encourage you to talk to other students regarding homework but you should not collaborate to the extent that two submissions are copies of each other. If you are found copying an assignment (from another student or internet), or if your submission has *unreasonable similarity* to another submission, you get a zero for that assignment automatically and this will get

reported to the University. I may ask you to come to my office and explain your code to me; in case you are not able to explain the code to my satisfaction, I'll assign you a zero in that project.

Instructional Strategies Our classes will mainly be lectures and question/answer sessions on the projects, with me using pdf files as notes. If you are not following something in the lectures, please stop me and ask a question. If you feel uncomfortable asking in the moment, please send me an email or comment after the class. Please do this, I encourage you asking questions. If a project is not clear to you, you should ask questions in class to get a clarification or send an email. If you do not have any idea on how to start working on a project, say so and I'll try my best to help you. However, I'll expect that you will ask project-related questions early on and not at the last minute when the project is due.

Miscellaneous If you have any disability that requires an accommodation (as per UMSL policy), you must notify me in advance. I'll try my best to make accommodation for you in that case.

You will have an account on the Linux machine `opsys.cs.ums1.edu` in the CS department and you should use it for all assignments. All programs must be done in C or C++. All projects must run on `opsys`, so if you code it in another environment you must test it on our server. You can access `opsys` from home by signing on to campus VPN, or by logging on to `delmar` and then, ssh'ing from there. The lecture notes will be available in PDF format. You may want to print the lecture notes before you come to class.

UMSL Course Policies

1. Participation (expectations)

- It is vitally important that our classroom environment promote the respectful exchange of ideas. This entails being sensitive to the views and beliefs expressed during discussions whether in class or online. Please speak with me before recording any class activity. It is a violation of University of Missouri policy to distribute such recordings without my authorization and the permission of others who are recorded.
- Your success in this course will heavily depend on your ability to communicate, engage and participate in all course activities. Successful completion of this course requires that a student keep up with all assignments, quizzes, projects and tests.
- Turn off sound on beepers, cell phones, and other devices during class. Adherence to the Student Conduct Code is expected.
- I am committed to insuring a positive learning environment by respecting that University policy (p. 66): <http://www.ums1.edu/~studentplanner/index.html>.
- If you are unable to participate in the scheduled class activities, you must notify the instructor within the week of that class module. **An unexcused failure to engage or participate with the class will be counted as an absence; unexcused absences may result in failure.** The instructor reserves the right to make judgment to accept and/or makeup assignments missed because of failed participation in the course activities.

2. Academic Integrity/Plagiarism

- Students are responsible for being attentive to and observant of campus policies about academic honesty as stated in the University's Student Conduct Code (p. 60): <http://www.ums1.edu/~studentplanner/index.html>
- To avoid accusations of academic dishonesty, please submit all written work to the Turnitin system before finalizing what you submit for evaluation. Check information about The Writing Center @UMSL that is linked to MyGateway Home.

- Plagiarism is the use of another persons words or ideas without crediting that person. Plagiarism and cheating will not be tolerated and may lead to failure on an assignment, in the class, and dismissal from the University. View this campus policy here: <http://www.ums1.edu/services/academic/policy/academic-dishonesty.html>
3. **Mandatory Reporting:** Under Title IX, all UMSL faculty, staff, and administrators (with limited exception) are obligated to report any incidents of sexual harassment, sexual misconduct, sexual assault, or gender discrimination to the Student Affairs office and/or other University officials. This ensures that all parties are protected from further abuses and that victim(s) are supported by trained counselors and professionals. Note: There are several offices at UMSL (e.g., Counseling Services, Health Services, Community Psychological Service, Center for Trauma Recovery, and Student Social Services) whose staff are exempt from Title IX mandated reporting, when the information is learned in the course of a confidential communication.
4. **Access, Disability and Communication**
- Students who have a health condition or disability, which may require accommodations in order to participate effectively in this course, should contact the **Disability Access Services Office**. Information about your disability is confidential.
 - 144 Millennium Student Center
 - Phone: (314) 516-6554
 - Website: <http://www.ums1.edu/services/disability/>
 - If you have difficulty communicating in English with the instructor of this course, contact the **Office of International Students and Scholar Services**:
 - Phone: (314) 516-5229
 - Email: iss@ums1.edu
 - Website: <http://www.ums1.edu/~intelstu/index.html>

Student Support and Services

- **Technical Support**
 - My Gateway (Blackboard): If you have problems logging into your online course, or an issue within the course site, please contact the **Technology Support Center**:
 - * Phone: (314) 516-6034
 - * Email: helpdesk@ums1.edu
 - * Website: <http://www.ums1.edu/technology/tsc/>
- **Academic Support**
 - **The Online Writing Lab:** At our My Gateway site, students can send their papers to our tutors, who will read them and send them back with suggestions. Students can also access SafeAssign, which identifies quoted material in their essays.
 - * Visit the online Writing Lab page on MyGateway to submit drafts online.
 - * We try to respond within 48 hours, but it may take longer, so allow ample time.
 - **NetTutor:** Online tutoring in many subjects is now available through NetTutor. In your courses on MyGateway, click on Tools and select NetTutor to log in.

- Student Services
 - The Student Retention Services office offers assistance tailored to specific student needs.
 - * 225 Millennium Student Center
 - * Phone: (314) 516-5300
 - * Email: umslsrs@umsl.edu
 - * Website: <http://www.umsl.edu/services/srs/>
- Departmental Tutoring
 - The department offers tutoring for up to Cmp Sci 3130, and occasionally for other courses
 - Check MyGateway organization CSTutoring.

Tentative Calendar

Week 1	Overview of operating systems and concepts
Week 2	Overview of operating systems and concepts; Interprocess Communications
Week 3	Interprocess Communications
Week 4	Interprocess Communications
Week 5	Interprocess Communications; Test 1
Week 6	Process Management
Week 7	Process Management
Week 8	Process Management
Week 9	Resource Management; Test 2
Week 10	Spring Break
Week 11	Resource Management
Week 12	Memory Management
Week 13	Memory Management
Week 14	File Systems
Week 15	File Systems; I/O Management
Week 16	I/O Management; Test 3

Detailed course outline

1. Introduction

- Basic elements of a computer system: processor, memory, I/O modules
- Evolution of a microprocessor
- Instruction execution
- Interrupts
- Processor registers
- Memory hierarchy
- Multiprocessor and multicore organization
- Classification of operating systems
- Process execution modes in Unix/Linux
- Overview of operating systems structuring methods like monolithic, layered, and object-oriented
- I/O communication
- Multiprogramming and multitasking

2. Process Coordination and Synchronization

- Process concept
- Overview of multitasking and role of interrupts
- Concurrent execution and precedence graphs (process flow diagrams)
- Race conditions, Sharing access
- Critical Section Problem
 - Models for two processes, including Dekker's Algorithm
 - Models for multiple processes, including Bakery Algorithm
 - Semaphores, mutex locks, and monitors

3. Process Scheduling and Dispatch

- Preemptive and non-preemptive scheduling strategies
- Short term, medium-term, and long-term scheduling

4. Deadlocks – Causes, conditions, avoidance, prevention, and resolution

5. Physical and Virtual Memory Organization

- Offset registers, partitions, pages, segments, swapping, overlays
- Caching and associative buffers
- Fetch, placement, and replacement policies
- Thrashing and working sets

6. Device and File Management

- File systems

- File types
- File access and directory management
- Distributed file systems
- Equivalence of files and devices
- Organization of I/O subsystems, free space management, and file layout
- Disk scheduling algorithms