ECE 465-565: Operating Systems Design
Fall 2024
Mon-Wed 3:00-4:15pm, 136 MRC

Course Syllabus

Instructor: Michela Becchi: mbecchi@ncsu.edu (office hours posted on course calendar)

Teaching Assistant: Mostafa Eghbali: meghbal@ncsu.edu (office hours posted on course calendar)

Course Overview

The course explores basic concepts and mechanisms related to the design of modern operating systems, including: process scheduling and coordination, memory management, synchronization, storage, file systems, security and protection, and their application to multi-core and many-core processors. *The course involves coding projects requiring strong C programming skills.*

Prerequisites

ECE465: ECE306 or CSC246; ECE309

ECE565: ECE306 or CSC246 or equivalent; ECE309 or equivalent

Notes: (1) *Strong C programming skills are required*; (2) ECE465, ECE565 and CSC501 are mutually exclusive: students may not receive credit for both ECE465 and ECE565, or both ECE465 and CSC501, or both ECE565 and CSC501.

Student Learning Outcomes

By the end of the course, students should be able to do the following:

1. Explain the concepts of processes, threads and their differences; apply this knowledge to design and implement multi-process and multi-threaded applications.
2. Explain the operation of different process scheduling algorithms and their strengths and weaknesses in terms of throughput, response time, CPU utilization and fairness.
3. Explain the concepts of race condition and critical section, and the use of different hardware and software synchronization mechanisms to guarantee mutual exclusion. These include: interrupt disabling, hardware synchronization mechanisms, mutexes, semaphores and condition variables.
4. Apply different synchronization mechanisms to classic synchronization problems, including: bounded-buffer, readers-writers and dining-philosopher problems.
5. Explain the concepts of virtual memory and address translation. Describe different address translation mechanisms (hardware- and software-based relocation, segmentation, paging) and memory allocation strategies (best fit, first fit, worst fit, slab and buddy allocators).
6. Describe the physical structure of secondary storage devices and its implications on the use of these devices and their performance; evaluate different disk scheduling algorithms.
7. Explain the function of a file system, its interfaces, and implementations; this includes evaluating advantages and disadvantages of different file system designs.

8. Design, implement, and validate basic scheduling, synchronization and memory management algorithms within an operating system.

ECE565 specific learning outcomes

9. Explain the interaction of architecture and operating system design.

10. Describe the implications of multi- and many-core systems on operating system design.

Textbooks and Other Course Material

Required textbook

□ R. and A. Arpaci-Dusseau - *Operating Systems: Three Easy Pieces*, Arpaci-Dusseau Books, August 2018 (Version 1.00) - http://pages.cs.wisc.edu/~remzi/OSTEP/

□ **Note**: An electronic edition of this textbook is freely available online. Paper/Kindle editions of the textbook are available for purchase (please see http://pages.cs.wisc.edu/~remzi/OSTEP/ for more information).

Additional resources (not required)

□ A. Silberschatz, P. Galvin, G. Gagne. *Operating System Concepts*, John Whiley & Sons, 2013. (9th edition)

□ W. Stallings. *Operating Systems – Internals and Design Principles*, Pearson, 2015. (8th edition)

Textbook covering the Xinu operating system (not required)

D. Comer. *Operating System Design – The Xinu Approach*, CRC Press, 2015 (2nd edition)

Course Schedule

A tentative course schedule (subject to change) is below:

Week	Date	Topic	Readings (from textbook)	Deadline (tentative)
Week 1	Aug 19	Introduction		
	Aug 21	OS structure	Section 2	
Week 2	Aug 26	Processes	Section 4-5	
	Aug 28	Processes		
Week 3	Sep 2	*Labor day (no classes)*		
	Sep 4	Threads	Section 26-27	
Week 4	Sep 9	*Overflow*		Project #1 due
	Sep 11	CPU scheduling	Section 7-8	
Week 5	Sep 16	CPU scheduling	Section 9-10	
	Sep 18	*Overflow*		
Week 6	Sep 23	Synchronization	Sections 28-29	Homework #1 due
	Sep 25	Synchronization	Section 30-31	
Week 7	Sep 30	Synchronization		
	Oct 2	*Overflow*		Project #2 due
Week 8	Oct 7	***Midterm exam***		
	Oct 9	Concurrency bugs	Section 32	
Week 9	Oct 14	*Fall break (no classes)*		
	Oct 16	Virtual memory – Introduction and Address spaces	Section 13	

Week 10	Oct 21	Virtual Memory – Address translation & Relocation	Section 15	
	Oct 23	Virtual Memory – Segmentation and free memory management	Section 16-17	
Week 11	Oct 28	Virtual Memory – Intro to paging and TLBs	Section 18-19	
	Oct 30	Virtual memory – Advanced Page tables	Section 20	Project #3 due
Week 12	Nov 4	Swapping	Section 21-22	
	Nov 6	*Overflow*		
Week 13	Nov 11	I/O Devices and hard disk drives	Section 36-37	
	Nov 13	RAID & file system interface	Section 38-39	
Week 14	Nov 18	File system implementation and FFS	Section 40-41	Homework #2 due
	Nov 20	File system – fsdk and journaling	Section 42	
Weak 15	Nov 25	File system – log-structured file system	Section 43	Project #4 due
	Nov 27	*Thanksgiving break*		
Weak 16	Dec 2	File system – SSD & NVM	Section 44	
	Dec 11	***Final exam (3:30-5:30PM)***		

Course structure

The bi-weekly lectures will cover the operating system concepts listed above, and students will be required to read textbook sections covering such concepts. All students will be required to perform programming assignments using XINU (https://xinu.cs.purdue.edu), an open-source operating system designed and developed by Prof. Douglas Comer's group at Purdue University. This hands-on experience is aimed to provide a more concrete understanding of operating systems design and of the interplays among different operating systems components. The instructor will provide homework assignments intended for exam preparation. These assignments will be graded only for students taking the course at the undergraduate level. Finally, there might be occasional quizzes to encourage students to keep up with the material and verify their high-level understanding of the concepts covered in class.

Exams

There will be two in-person exams. The midterm exam, accounting for 15% of the grade, is scheduled around the midpoint of the semester during class time (October 7, 3:00PM). The final exam, accounting for 30% of the grade, will be given during the final exam period (December 11, 3:30PM). The final exam will be a ***cumulative*** 120-minute exam. Exams can include questions on the course projects. For both exams, you can use a one-page cheat sheet written/typed on both sides. No calculator is allowed. *The complexity of the exams will be higher for students taking the course at the graduate level (e.g., additional or more involved problems).*

Projects

There will be several (tentatively, 4) projects (programming assignments). These projects require strong C programming skills and will be performed individually (unless otherwise specified). For projects performed in teams, both team members are expected to fully understand and be able to explain and reproduce any aspects of the project, independently of how they have planned the work in the team. Quizzes and exams might include questions related to the projects. Students should indicate in their report any resources used other than the textbook. *Reusing (even partially) code from other students or online resources is strictly forbidden and considered academic* *dishonesty. Although projects are required for all students, their complexity will be lower for students taking the course at the undergraduate level.*

Homework assignments

The instructor will assign some (tentatively, 2) homework assignments. As indicated above, *these assignments will be graded only for students taking the course at the undergraduate level*. Student taking the course at the graduate level are *highly encouraged* to work on the assignments, as they are intended as a tool for exam preparation.

Quizzes

Students are expected to attend class and participate in discussions. Class participation will be measured through pop quizzes using TopHat (https://tophat.com/).

Grading (subject to adjustments)

The overall grade will be a weighted average of the following components:

Students taking the course at the undergraduate level (ECE465)

- Exams (45%) – two exams (15% midterm/30% final)
- Projects (40%) – 4 programming assignments (weights TBD)
- Homework assignments (10%) – 2 homework assignments (weights TBD)
- Quizzes (5%)

Students taking the course at the graduate level (ECE565)

- Exams (45%) – two exams (15% midterm/30% final)
- Projects (50%) – 4 programming assignments (weights TBD)
- Quizzes (5%)

The total course grade is a weighted average, with the weights described above. I guarantee the following assignment of grades based on the weighted average:

- 98-100: A+
- 93-97: A
- 90-92: A-
- 87-89: B+
- 82-86: B
- 79-81: B-
- 76-78: C+
- 71-75: C
- 68-70: C-
- 60-67: D
- < 60: F

I reserve the right to shift the numerical cutoff points down (but never up), based on overall class performance, problems with a particular assignment, etc.

Use of ChatGPT

The course assignments are designed not to require any additional information other than the textbook, the lecture notes and the Intel Architecture Software Development's Manual (posted on Moodle). However, while not recommended, the use of ChatGPT (https://chat.openai.com/) is permitted in order to integrate the knowledge acquired in the class and facilitate learning. To that end, the following restrictions apply:

- ChatGPT is not an alternative to the textbook. The textbook is a more comprehensive and reliable source of information.
- It is not allowed to use ChatGPT to obtain code pertaining the projects.
- It is not allowed to use ChatGPT to answer homework questions.
- The questions entered in ChatGPT must be listed and submitted using the form below.

In other words: projects and assignments should reflect the student's understanding of the material and coding abilities. Here's a rule of thumb: you are allowed to ask ChatGPT questions that you would feel comfortable asking the instructor. *In case of doubt, please consult the instructor.*

Example of questions allowed:

- *What is the syntax of typedef in the C language?*
- *How does a struct work in C?*
- *What are the registers of a x86 processor?*
- *What are the stack conventions for a x86 processor?*
- *What are the CR1, CR2 and CR3 registers for?*

Example of questions not allowed:

- *Any questions containing the word: "Xinu".*
- *Show me an example of code implementing a lock.*
- *Show me an example of code implementing the fork system call.*

Important: if you used ChatGPT to get help with a project, you should list all the questions that you have entered in ChatGPT using the following form: https://forms.gle/vnKSTPPp64Hx99Ro8. The form must be submitted together with the corresponding project. Note: this form is restricted to NC State – you must log in using your NC State account to view it.

Class Policies

Late work – A 3-day delay will be allowed on both homework assignments and projects, *with 5% grade deduction for each day of delay*. If you have a *certified* medical excuse or instructor approval, you may receive full credit if the assignment is turned in as soon as possible. *A missed exam can only be made up in the case of a university-excused absence*.

Attendance – While I won't take attendance, students should attend all lectures. *No makeups will be given for missed discussions, quizzes, etc. given during class time.*

Credit-only requirements – If you are taking this course for credit-only, your grade must be equivalent to C- or better to receive an S (Satisfactory) grade. Otherwise, you will receive a U (Unsatisfactory) grade.

Audit requirements – If you are auditing this course, you must only perform the midterm and final exam. You should score at least 70% in both exams to receive an AU (Audit) grade. Otherwise, you will receive a NR (No Recognition) grade.

Information for EOL students

- Projects will be submitted through Moodle. Graded material will be returned within 2 weeks from submission. Grades will be posted in Moodle.
- Besides regular office hours, additional office hours can be scheduled by appointment.
- Exams will be held according to the calendar above and will be coordinated by the EOL staff. More information will be provided over the course of the semester.
- Quizzes (and their due date) will be announced via email.

Computer Resources

Course page: Moodle: https://moodle-courses2425.wolfware.ncsu.edu/course/view.php?id=2776

The course Moodle page will contain the syllabus, lecture notes, projects and homework assignments, and other relevant information.

Course mailing list: Piazza: see link in Moodle

All class announcements will be posted in Piazza.

I encourage students to use Piazza to post *questions related to the course material, homework assignments, and projects*, and to post answers to other students' questions. Please note that Piazza will be used for announcements and for clarifications on the course material. *If inappropriate material (such as requests of extensions or grade changes, or offensive content) is posted, the message will be deleted at the instructor's discretion*.

What kind of information can I post to answer questions posed by other students? If the question concerns a project or homework assignment, you are **not** allowed to post code (or the solution to the problem's posed). You can post information helping the student to find an answer to the question. In other words, any information provided on Piazza should help other students to understand or to interpret a problem/question, but each student should then independently come up with his/her own solution.

My preferred mode of communication is Piazza, followed by email. *If you email me directly, please prepend "[ECE465]" or "[ECE565]" (whatever applies) to the subject of your email*. Try to reserve email for questions that require privacy.

Academic Integrity

Students are required to comply with the university policy on academic integrity found in the Code of Student Conduct found at http://policies.ncsu.edu/policy/pol-11-35-01.

All exams, quizzes, projects and homework assignments are individual assignments, unless otherwise stated in writing. As mentioned above, in case of projects and homework assignments performed in teams, both team members must be able to explain all parts of the assignments. *Not being able to explain a homework assignment/project submitted is considered cheating.*

This course has a ***strict no-code reuse policy***. Under no circumstances is it allowed to ***consult or reuse*** code from other individuals or online resources. Code reuse (**even if partial**) from other students, online or other resources will be considered an academic integrity violation. In case of doubts, please ask the instructor in advance.

Evidence of cheating, plagiarism, or other violations of the Code of Student Conduct will be investigated and, if appropriate, referred to the Office of Student Conduct for disciplinary review. When an academic integrity violation is detected, the student will be reported to the Office of Student Conduct and will receive a 0 score in the related exam, project or assignment. In the presence of multiple academic integrity violations by the same student, the student will fail the class (in addition to be reported to the Office of Student Conduct).

The Code of Student Conduct can be found at: http://policies.ncsu.edu/policy/pol-11-35-01.

Inclement Weather

The class will follow the University's closure policy. If you are wondering whether classes are cancelled due to inclement weather, please check the University website or the weather hotline (513-8888).

Statement for Students with Disabilities

Reasonable accommodations will be made for students with verifiable disabilities. In order to take advantage of available accommodations, students must register with the Disability Resource Office at Holmes Hall, Suite 304, Campus Box 7509, 919-515-7653. For more information on NC State's policy on working with students with disabilities, please see the Academic Accommodations for Students with Disabilities Regulation (REG02.20.01) (https://policies.ncsu.edu/regulation/reg-02-20-01/).

Non-Discrimination Policy

NC State University provides equality of opportunity in education and employment for all students and employees. Accordingly, NC State affirms its commitment to maintain a work environment for all employees and an academic environment for all students that is free from all forms of discrimination. Discrimination based on race, color, religion, creed, sex, national origin, age, disability, veteran status, or sexual orientation is a violation of state and federal law and/or NC State University policy and will not be tolerated. Harassment of any person (either in the form of quid pro quo or creation of a hostile environment) based on race, color, religion, creed, sex, national origin, age, disability, veteran status, or sexual orientation also is a violation of state and federal law and/or NC State University policy and will not be tolerated. Retaliation against any person who complains about discrimination is also prohibited. NC State's policies and regulations covering discrimination, harassment, and retaliation may be accessed at http://policies.ncsu.edu/policy/pol-04-25-05 or http://www.ncsu.edu/equal_op/. Any person who feels that he or she has been the subject of prohibited discrimination, harassment, or retaliation should contact the Office for Equal Opportunity (OEO) at 919-515-3148.

N.C. State University Policies, Regulations, and Rules (PRR)

Students are responsible for reviewing the PRRs that pertain to their course rights and responsibilities. These include:

□ http://policies.ncsu.edu/policy/pol-04-25-05(Equal Opportunity and Non-Discrimination Policy Statement)
□ http://oied.ncsu.edu/oied/policies.php (Office for Institutional Equity and Diversity)
□ http://policies.ncsu.edu/policy/pol-11-35-01 (Code of Student Conduct)
□ http://policies.ncsu.edu/regulation/reg-02-50-03 (Grades and Grade Point Average).