Course Syllabus

Jump to Today ✎ Edit

CS 332: Principles of Operating Systems - Honors, Spring 2025

Course Overview

The principles underlying operating systems. Topics include process management, thread concurrency, CPU scheduling, synchronization, memory management, file systems, and virtualization.

Instructor: Prof. Zhihao “Zephyr” Yao, Ph.D.

Email: zhihao.yao@njit.edu

Office Hours: Monday 11:00 AM - 12:00 PM, GITC 4317A

Class Schedule: MW 1:00 PM - 2:20 PM, CKB 303

Teaching Assistant: Glenn Merritt (gmm46@njit.edu (mailto:gmm46@njit.edu))

TA's Office Hours: By appointment

Course Textbook:

Required reading:
Operating System Concepts, 10th Edition
ISBN: 978-1-119-32091-3
Authors: Abraham Silberschatz, Peter B. Galvin, Greg Gagne

Optional reading:
Linux in a Nutshell: A Desktop Quick Reference (6th Edition)
ISBN: 978-0596154486
Authors: Ellen Siever, Stephen Figgins, Robert Love, Arnold Robbins

Linux Kernel Development (3rd Edition)

ISBN: 978-0672329463
Authors: Robert Love

Operating Systems: Three Easy Pieces
ISBN: 978-1985086593
Authors: Remzi H Arpaci-Dusseau, Andrea C Arpaci-Dusseau

Other resources:
CS 332 Syllabus from previous semester [1]

Assessment and Grading

Standard grading matrix:

Midterm Exam 1	20%
Midterm Exam 2	20%
Final Exam	20%
Project	20%
Discussion Lead	20%
Extra Credit (Course Evaluation)	1%

Letter Grades

Letter Grades will be assigned in accordance with the NJIT undergraduate grade legend (**https://www.njit.edu/dos/policies/gradingpolicy.php (https://www.njit.edu/dos/policies/gradingpolicy.php)**) converting numerical scores to letter grades. A curve is not guaranteed and should not be expected. However, the instructor reserves the right to apply a curve at their discretion if deemed necessary to ensure fair assessment.

- A: 90.00% and above
- B+: 85.00% and above
- B: 80.00% and above
- C+: 75.00% and above
- C: 70.00% and above
- D: 60.00% and above
- F: Below 60.00%

Exams: All exams are closed-book, and 1 hour 20 minutes. Midterm 1: Focuses on the material covered from the beginning of the course up to the exam date. Midterm 2: Covers content discussed after Midterm 1 and prior to the second exam. Final Exam: Comprehensive, covers the entire course content. All exams are conducted in-person, except for (1) individual cases approved by the Dean of

Student, (2) an official campus closure due to extreme weather or other circumstances.

Discussion Lead:

Leading discussion accounts for 20% of your total grade. Each student will lead a 20-minute discussion on an assigned article or reading related to the week's chapter. These assigned readings are research articles that complement the course material.

Sign-Up Process:

The discussion schedule operates on a first-come, first-serve basis. Students must sign up for the week of their choice at the start of the semester. The selection deadline is Monday, January 27th, 11:59PM EST. The sign up must be done using the Canvas Assignment "Discussion Sign Up". Each week's assigned article is announced in the course schedule below.

After the sign-up deadline, students are allowed to switch their assigned week to lead the discussion with another student under the following conditions: (1) Both students involved in the switch must agree to the change. (2) The instructor must be informed of the switch at least one day prior to the class session of the affected discussion. (3) It is the responsibility of the student originally assigned to the week to ensure that the instructor is notified and the change is confirmed. (4) Failure to Notify: If the instructor is not informed in time, the originally assigned student will remain responsible for leading the discussion.

Expectations for the Discussion Lead:

- Provide a concise summary (slides are strongly encouraged, but using hand-writing or presenting the original documents are acceptable) of the article, including its main points, methodology, and key findings or key concepts.
- Prepare 2-3 discussion questions to engage the class and encourage participation.
- Each item will be graded on a scale of 1-4, with 4 being excellent and 1 being unsatisfactory.

Rubrics for the Discussion Lead:

Criteria	Description	Scale (1-4)
Preparation and Organization	Presentation demonstrates clear preparation, with organized content, logical flow.	1 = Poor, 2 = Fair, 3 = Good, 4 = Excellent
Content Mastery	Presenter demonstrates a deep understanding of the article, be able to accurately summarizes key points.	1 = Poor, 2 = Fair, 3 = Good, 4 = Excellent

Engagement	Facilitates an engaging discussion by asking questions, encouraging participation, and managing the flow of discussion effectively.	1 = Poor, 2 = Fair, 3 = Good, 4 = Excellent
Clarity	Communicates ideas clearly, and uses visuals or examples effectively to support the presentation.	1 = Poor, 2 = Fair, 3 = Good, 4 = Excellent
Relevance and Insight	Connects the article to course material and demonstrates insight into broader implications or applications.	1 = Poor, 2 = Fair, 3 = Good, 4 = Excellent

Course Projects:

A team can have at most three people. Each of team members must contribute. Reports need to specify the contributions of each team member. At any point, a team may be split, but no new teams are allowed to form after the team name and formation due date. After a team is split, the separated members are allowed to use the team's work prior to the split.

Team name and formation due 1/27 11:59 PM EST
First progress report due 2/24 11:59 PM EST
Second progress report due 3/26 11:59 PM EST
Final report, code, and testing results (as a zip file) due 5/7 11:59 PM EST

Policies

Accommodations and Supports

If you need an accommodation due to a disability please contact the Office of Accessibility Resources and Services at OARS@NJIT.EDU, or visit us in Kupfrian Hall 201 to discuss your specific needs. A Letter of Accommodation Eligibility from the office authorizing student accommodations is required.

Deadlines

- All deadlines are firm (in EST).
- Late project submissions will be accepted up to 50 hours past the deadline, with a 2% deduction for each hour late.

Absenteeism

- You are responsible for catching up on any material or information missed if you do not attend

class.

- If you miss one exam due to special circumstances, you must contact the Dean of Students (DOS) within 2 working days from the day the reason for the absence is lifted with all necessary documentation. If DOS approves, your missing exam grade will be set equal to the average of the non-missing exam grades.
- Missing two exams leads to an automatic F in the course.

Incomplete

A grade of I (incomplete) is given only in **rare** circumstances when a student would normally have completed the course work, but could not do so because of documented special circumstances. See NJIT Catalog for details: **https://catalog.njit.edu/undergraduate/academic-policies-procedures (https://catalog.njit.edu/undergraduate/academic-policies-procedures,)**

Exceptions to Policies

- I have strict policies regarding deadlines and absenteeism. But, exceptions to these policies include documented medical and other officially excusable absence determined by the Dean of Students (DOS): **https://www.njit.edu/dos/student-absence-verification (https://www.njit.edu/dos/student-absence-verification)**
- You must contact the Dean of Students (DOS) within 2 working days from the day the reason for the absence is lifted with all necessary documentation.

Use of ChatGPT and other AI Writing or Coding Tools in Course Project

- The use of AI tools, such as ChatGPT, in course projects is allowed but must adhere to the following guidelines:
  - Students are fully responsible for the accuracy and correctness of all content in their submissions, including any text or code generated by AI. If AI-generated content is incorrect, the assignment will be graded accordingly. Note that AI is known to output incorrect statements and codes.
  - Students must disclose their use of AI tools in a visible part of the assignment, such as a code comment, footnote, or similar section. This disclosure is required for each graph, table, paragraph, or function where AI was used. The disclosure must clearly identify the AI tool and version used and describe any post-generation edits made.
  - Using a single set of prompts to generate an entire coding project or report is strictly prohibited. These attempts are easily detectable by both human graders and software tools and will result in a grade of zero for the submission.

Course Schedule

Week	Date	Topic	Discussion
1	Jan 22 (Wed)	Introduction & Chapter 1 - Computer Organization	
2	Jan 27 (Mon)	Chapter 1 - Computer Organization (cont.)	
	Jan 29 (Wed)	Chapter 2 - Operating-System Structures	
3	Feb 3 (Mon)	Chapter 3 - Processes	CTSS Technical Notes (Chapter 1) [2]
	Feb 5 (Wed)	Chapter 3 - Processes (cont.)	
4	Feb 10 (Mon)	Chapter 4 - Threads and Concurrency	Linux scheduler [4]
	Feb 12 (Wed)	Chapter 4 - Threads and Concurrency (cont.)	
5	Feb 17 (Mon)	Chapter 5 - CPU Scheduling	Serverless Computing [12]
	Feb 19 (Wed)	Chapter 5 - CPU Scheduling (cont.)	
6	Feb 24 (Mon)	Chapter 6 - Synchronization Tools	Race Conditions [10]
	Feb 26 (Wed)	**Midterm 1** (Chapters 1-6)	
7	Mar 3 (Mon)	Chapter 6 - Synchronization Tools (cont.)	Android IPC [11]
	Mar 5 (Wed)	Chapter 7 - Synchronization Examples	
8	Mar 10 (Mon)	Chapter 7 - Synchronization Examples (cont.)	Microkernels [3]
	Mar 12 (Wed)	Chapter 8 - Deadlocks	
	Mar 16 - Mar 22	**Spring Recess – No Classes**	
9	Mar 24 (Mon)	Chapter 8 - Deadlocks (cont.)	OS deadlock [16]
	Mar 26 (Wed)	Chapter 9 - Main Memory	Memory bugs [9]
10	Mar 31 (Mon)	Chapter 9 - Main Memory (cont.)	Memory bugs in AI-written code [17]
	Apr 2 (Wed)	Chapter 10 - Virtual Memory	
11	Apr 7 (Mon)	Chapter 10 - Virtual Memory (cont.)	KASLR [15]
	Apr 9 (Wed)	Chapter 12 - I/O System	
12	Apr 14 (Mon)	Chapter 12 - I/O System (cont.)	Trusted I/O [7]

	Apr 16 (Wed)	**Midterm 2** (Chapters 7-12)	
13	Apr 21 (Mon)	Chapter 13 - File-System Interface	RAID [8]
	Apr 23 (Wed)	Chapter 13 - File-System Interface (cont.)	
14	Apr 28 (Mon)	Chapter 18 - Virtual Machine	Graphics virtualization [5]
	Apr 30 (Wed)	Chapter 18 - Virtual Machine (cont.)	
15	May 5 (Mon)	Mobile and Cloud Operating Systems [6, 13, 14]	Mobile availability [13]
	May 7 (Wed)	**No Class (Friday Classes Meet Per University Calendar)**	
16	Final Exam Week	**Final Exam (Comprehensive)** (Exact date TBD)	

Learning Outcomes

By the end of this course, students will be able to:

1. Describe the fundamental components of an operating system and their functions.
2. Explain process management and apply concepts of concurrency and synchronization.
3. Analyze and implement different CPU scheduling algorithms.
4. Understand virtual memory and its hardware support.
5. Understand the role of CPU in privilege management, interrupts, and I/O systems.

6. Explain and simulate solutions to avoid deadlocks in operating systems.
7. Understand file systems interfaces, and the abstraction for storage devices.
8. Explain virtualization methods in modern computing.

Academic Integrity:

Academic Integrity is the cornerstone of higher education and is central to the ideals of this course and the university. Cheating is strictly prohibited and devalues the degree that you are working on. As a member of the NJIT community, it is your responsibility to protect your educational investment by knowing and following the academic code of integrity policy that is found at: ***NJIT Academic Integrity Code*** **(https://t.e2ma.net/click/r146wkb/3i2x1wml/rhhrpwx)** *.*

Please note that it is my professional obligation and responsibility to report any academic misconduct to the Dean of Students Office. Any student found in violation of the code by cheating, plagiarizing or using any online software inappropriately will result in disciplinary action. This may include a failing grade of F, and/or suspension or dismissal from the university. If you have any questions about the code of Academic Integrity, please contact the Dean of Students Office at ***dos@njit.edu*** **(mailto:dos@njit.edu)**

This syllabus is subject to updates. See below for a history of syllabus revisions:

Syllabus Revisions:

Note that this syllabus is subject to updates. Any changes to the syllabus after the first day of class will be announced and recorded below.

History of syllabus revisions:

1/22/2025 Set team formation deadline to January 27th 11:59PM; update discussion article for Week 9, 10.

Reference:

[1] Yao, Z. (2024). CS 332-001: Principles of Operating Systems.

https://digitalcommons.njit.edu/cgi/viewcontent.cgi?article=1490&context=cs-syllabi
(https://digitalcommons.njit.edu/cgi/viewcontent.cgi?article=1490&context=cs-syllabi)

[2] Saltzer, J. H. (1965). CTSS technical notes.

https://dspace.mit.edu/bitstream/handle/1721.1/149338/MIT-LCS-TR-016.pdf
(https://dspace.mit.edu/bitstream/handle/1721.1/149338/MIT-LCS-TR-016.pdf)

[3] Elphinstone, K., & Heiser, G. (2013, November). From L3 to seL4 what have we learnt in 20 years of L4 microkernels?. In Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles (pp. 133-150).

https://dl.acm.org/doi/pdf/10.1145/2517349.2522720
(https://dl.acm.org/doi/pdf/10.1145/2517349.2522720)

[4] Lozi, J. P., Lepers, B., Funston, J., Gaud, F., Quéma, V., & Fedorova, A. (2016, April). The Linux scheduler: a decade of wasted cores. In Proceedings of the Eleventh European Conference on Computer Systems (pp. 1-16).

Computer Systems (pp. 1-16).

https://dl.acm.org/doi/pdf/10.1145/2901318.2901326 (https://dl.acm.org/doi/pdf/10.1145/2901318.2901326)

[5] Yao, Z., Ma, Z., Liu, Y., Amiri Sani, A., & Chandramowlishwaran, A. (2018). Sugar: Secure GPU acceleration in web browsers. ACM Sigplan Notices, 53(2), 519-534.

https://dl.acm.org/doi/pdf/10.1145/3296957.3173186 (https://dl.acm.org/doi/pdf/10.1145/3296957.3173186)

[6] Minimizing a Smartphone's TCB for Security-Critical Programs with Exclusively-Used, Physically-Isolated, Statically-Partitioned Hardware

https://dl.acm.org/doi/pdf/10.1145/3581791.3596864 (https://dl.acm.org/doi/pdf/10.1145/3581791.3596864)

[7] Liu, Y., Yao, Z., Chen, M., Amiri Sani, A., Agarwal, S., & Tsudik, G. (2024, May). ProvCam: A Camera Module with Self-Contained TCB for Producing Verifiable Videos. In Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (pp. 588-602).

https://dl.acm.org/doi/pdf/10.1145/3636534.3649383 (https://dl.acm.org/doi/pdf/10.1145/3636534.3649383)

[8] Patterson, D. A., Gibson, G., & Katz, R. H. (1988, June). A case for redundant arrays of inexpensive disks (RAID). In Proceedings of the 1988 ACM SIGMOD international conference on Management of data (pp. 109-116).

https://dl.acm.org/doi/pdf/10.1145/50202.50214 (https://dl.acm.org/doi/pdf/10.1145/50202.50214)

[9] Martin, B., Brown, M., Paller, A., Kirby, D., & Christey, S. (2011). 2011 CWE/SANS top 25 most dangerous software errors. Common Weakness Enumeration, 7515.

https://www.open-std.org/JTC1/SC22/WG23/docs/ISO-IECJTC1-SC22-WG23_N0019-CWE-Overview-martin.pdf (https://www.open-std.org/JTC1/SC22/WG23/docs/ISO-IECJTC1-SC22-WG23_N0019-CWE-Overview-martin.pdf)

[10] Bishop, M., & Dilger, M. (1996). Checking for race conditions in file accesses. Computing systems, 2(2), 131-152.

https://www.usenix.org/legacy/publications/compsystems/1996/spr_bishop.pdf (https://www.usenix.org/legacy/publications/compsystems/1996/spr_bishop.pdf)

[11] Artenstein, N., & Revivo, I. (2014, September). Man in the binder: He who controls ipc, controls the droid. In Eur. BlackHat Conf.

https://sc1.checkpoint.com/downloads/Man-In-The-Binder-He-Who-Controls-IPC-Controls-The-Droid-wp.pdf (https://sc1.checkpoint.com/downloads/Man-In-The-Binder-He-Who-Controls-IPC-Controls-The-Droid-wp.pdf)

[12] Yu, H., Basu Roy, R., Fontenot, C., Tiwari, D., Li, J., Zhang, H., ... & Park, S. J. (2024, April). RainbowCake: Mitigating Cold-starts in Serverless with Layer-wise Container Caching and Sharing. In Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1 (pp. 335-350).

https://dl.acm.org/doi/pdf/10.1145/3617232.3624871 (https://dl.acm.org/doi/pdf/10.1145/3617232.3624871)

[13] Zueck, D., Atallah, N., Do, I., Yao, Z., & Sani, A. A. (2024, September). Hora: High Assurance Periodic Availability Guarantee for Life-Critical Applications on Smartphones. In Proceedings of the 15th ACM SIGOPS Asia-Pacific Workshop on Systems (pp. 115-121).

https://dl.acm.org/doi/pdf/10.1145/3678015.3680486 (https://dl.acm.org/doi/pdf/10.1145/3678015.3680486)

[14] Enck, W., Ongtang, M., & McDaniel, P. (2009). Understanding android security. IEEE security &

privacy, 7(1), 50-57.

https://dl.acm.org/doi/pdf/10.1145/3678015.3680486 (https://dl.acm.org/doi/pdf/10.1145/3678015.3680486)

[15] Stanley, D. M., Xu, D., & Spafford, E. H. (2013, December). Improved kernel security through memory layout randomization. In 2013 IEEE 32nd International Performance Computing and Communications Conference (IPCCC) (pp. 1-10). IEEE.

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6742768 (https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6742768)

[16] Habermann, A. N. (1969). Prevention of system deadlocks. Communications of the ACM, 12(7), 373-ff.

https://dl.acm.org/doi/pdf/10.1145/363156.363160 (https://dl.acm.org/doi/pdf/10.1145/363156.363160)

[17] Chong, C. J., Yao, Z., & Neamtiu, I. (2024). Artificial-Intelligence Generated Code Considered Harmful: A Road Map for Secure and High-Quality Code Generation. arXiv preprint arXiv:2409.19182.

https://arxiv.org/pdf/2409.19182 (https://arxiv.org/pdf/2409.19182)