

Syllabus: CS 6210 Adv. Operating Systems

Fall 2026

Instructor: Prof. Umakishore Ramachandran

Note: Mostly finalized, but subject to change. Students are responsible for regularly reviewing the syllabus for any updates or changes to the course schedule and requirements.

Course Schedule

Week	Activity	Video Length	Due Date / Window	Notes
1	OS Refresher (Optional)	2h 18m	Fri Aug 28	For students who need a refresher on AOS topics
1	Lesson 1: Intro to AOS	46m	Fri Aug 28	
2	Homework		Tue Sep 1 (11:59 PM ET)	1 week
2	Pre-lab		Tue Sep 1 (11:59 PM ET)	1 week
2	Lesson 2: OS Structures	2h 40m	Fri Sep 4	
3	Lesson 3: Virtualization	1h 53m	Fri Sep 11	
4–5	Lesson 4: Parallel Systems	5h 34m	Fri Sep 18 / Fri Sep 25	
6	Project 1: Virtual Machine Scheduling in KVM		Mon Sep 28 (11:59 PM ET)	3 weeks (available starting Mon Sep 7 at 11:59 PM ET)
6–7	Lesson 5: Distributed Systems	3h 20m	Fri Oct 2 / Fri Oct 9	
6–7	Test 1 (Lessons 1–4)		Fri Oct 2 (11:59 PM ET) – Mon Oct 5 (11:59 PM ET)	
	Fall Break for GT Campus		Mon Oct 5 – Tue Oct 6	Does not impact OMS
8	Lesson 6: Distributed Objects and Middleware	1h 56m	Fri Oct 16	
9	Project 2: Barrier Synchronization		Mon Oct 19 (11:59 PM ET)	3 weeks (available starting Mon Sep 28 at 11:59 PM ET)
9–10	Lesson 7: Distributed Subsystems	3h 48m	Fri Oct 23 / Fri Oct 30	
11	Lesson 9: Internet Computing	2h 34m	Fri Nov 6	
11	Project 3: Distributed Service using gRPC		Mon Nov 9 (11:59 PM ET)	3 weeks (available starting Mon Oct 19 at 11:59 PM ET)

Week	Activity	Video Length	Due Date / Window	Notes
12	Lesson 10: RT and Multimedia	1h 15m	Fri Nov 13	
12–13	Test 2 (Lessons 5–7)		Fri Nov 6 (11:59 PM ET) – Mon Nov 9 (11:59 PM ET)	
13	Lesson 8: Failures and Recovery	1h 58m	Fri Nov 20	
14	Lesson 11: Security	1h 17m	Fri Nov 27	
16	Project 4: Implement MapReduce Framework		Mon Dec 7 (11:59 PM ET)	4 weeks (available starting Mon Nov 3 at 11:59 PM ET)
17	Test 3 (Lessons 8–11)		Fri Dec 11 (11:59 PM ET) – Mon Dec 14 (11:59 PM ET)	

Reading List

Lecture	Papers
Lesson 2: OS Structures	1. Bershad et al., “Extensibility, Safety and Performance in the SPIN Operating System,” SOSP 1995. 2. Engler, Kaashoek, O’Toole, “Exokernel: An Operating System Architecture for Application-Level Resource Management,” SOSP 1995. 3. Liedtke, “On Microkernel Construction,” SOSP 1995. 4. Liedtke, “Improved Address-Space Switching on Pentium Processors by Transparently Multiplexing User Address Spaces,” GMD TR No. 933, Nov 1995 (<i>self-study</i>).
Lesson 3: Virtualization	1. Barham, Dragovic, Fraser, Hand, Harris, Ho, Neugebauer, Pratt, Warfield, “Xen and the Art of Virtualization,” SOSP 2003. 2. Waldspurger, “Memory Resource Management in VMware ESX Server,” OSDI 2002.

Lecture	Papers
Lesson 4: Parallel Systems	1. Mellor-Crummey & Scott, “Algorithms for Scalable Synchronization on Shared-Memory Multiprocessors,” TOCS, Feb 1991. 2. Bershad, Anderson, Lazowska, Levy, “Lightweight Remote Procedure Call,” TOCS 8(1):37–55, Feb 1990. 3. <i>(partial reading: skip system modeling)</i> Squillante & Lazowska, “Using Processor-Cache Affinity Information in Shared Memory Multiprocessor Scheduling,” IEEE TPDS, Feb 1993, 131–143. 4. Fedorova, Seltzer, Small, Nussbaum, “Performance of Multithreaded Chip Multiprocessors and Implications for Operating System Design,” USENIX 2005. 5. Gamsa, Krieger, Appavoo, Stumm, “Tornado: Maximizing Locality and Concurrency in a Shared Memory Multiprocessor OS,” OSDI 1999. 6. <i>(partial reading: Sec 1, 2, 3, 10)</i> Boyd-Wickizer et al., “Corey: An Operating System for Many Cores,” OSDI 2008. 7. <i>(partial reading: Sec 1, 2, 3, 8)</i> Govil, Teodosiu, Huang, Rosenblum, “Cellular Disco: Resource Management Using Virtual Clusters on Shared-Memory Multiprocessors,” SOSP 1999.
Lesson 5: Distributed Systems	1. Lamport, “Time, Clocks, and the Ordering of Events in a Distributed System,” CACM 21(7):558–565, July 1978. 2. Thekkath & Levy, “Limits to Low-Latency Communications on High-Speed Networks,” TOCS, May 1993. 3. Hutchinson & Peterson, “The x-Kernel: An Architecture for Implementing Network Protocols,” TSE 17(1):64–76, Jan 1991. 4. Wetherall, “Active Networks: Vision and Reality: Lessons from a Capsule-based System,” SOSP 1999 (OSR 33(5)). 5. Liu, Kreitz, van Renesse, Hickey, Hayden, Birman, Constable, “Building Reliable High Performance Communication Systems from Components,” SOSP 1999 (OSR 33(5)). 6. <i>(partial reading)</i> Schroeder & Burrows, “Performance of the Firefly RPC,” SOSP 1989.
Lesson 6: Distributed Objects and Middleware	1. Mitchell et al., “An Overview of the Spring System,” Compcon, Feb 1994. 2. Hamilton, Powell, Mitchell, “Subcontract: A Flexible Base for Distributed Programming,” SOSP 1993. 3. Wollrath, Riggs, Waldo, “A Distributed Object Model for the Java System,” USENIX COOTS, May 1996. 4. Cecchet, Marguerite, Zwaenepoel, “Performance and Scalability of EJB Applications,” OOPSLA.

Lecture	Papers
Lesson 7: Distributed Subsystems	1. Feeley, Morgan, Pighin, Karlin, Levy, Thekkath, “Implementing Global Memory Management in a Workstation Cluster,” SOSP 1995. 2. Amza, Cox, Dwarkadas, Keleher, Lu, Rajamony, Yu, Zwaenepoel, “TreadMarks: Shared Memory Computing on Networks of Workstations,” IEEE Computer, Feb 1996. 3. Anderson et al., “Serverless Network File System,” <i>ACM Transactions on Computer Systems</i>, Feb 1996. 4. (<i>partial reading</i>) Satyanarayanan, “Coda: A Highly Available File System for a Distributed Workstation Environment,” IEEE Trans. Computers, Apr 1990.
Lesson 8: Failures and Recovery	1. Satyanarayanan et al., “Lightweight Recoverable Virtual Memory,” SOSP 1993, 146–160. 2. Lowell & Chen, “Free Transactions with Rio Vista,” SOSP 1997. 3. Haskin et al., “Recovery Management in QuickSilver,” TOCS, Feb 1988. 4. (<i>read on your own</i>) Gray et al., “The Recovery Manager of a Data Management System,” ACM Computing Surveys 13(2), June 1981, 223–242. 5. (<i>partial reading: first 3 sections</i>) Porter, Hofmann, Rossbach, Benn, Witchel, “Operating System Transactions,” SOSP 2009. 6. (<i>partial reading</i>) Peng & Dabek, “Large-scale Incremental Processing Using Distributed Transactions and Notifications,” OSDI 2010.
Lesson 9: Internet Computing	1. Dean & Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters.” 2. (<i>partial reading</i>) Brewer, “Lessons from Giant-Scale Services.” 3. (<i>partial reading</i>) Barroso, Dean, Hölzle, “Web Search for a Planet: The Google Cluster Architecture,” IEEE Micro. 4. Freedman, Freudenthal, Mazières, “Democratizing Content Publication with Coral.” 5. DeCandia et al., “Dynamo: Amazon’s Highly Available Key-value Store,” SOSP 2007. 6. (<i>Read on your own – Web technologies</i>) (a) Curbera et al., “Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI,” IEEE Internet Computing 6(2), 2002, 86–93. (b) Curbera, Khalaf, Mukhi, Tai, Weerawarana, “The Next Step in Web Services,” CACM 46(10), Oct 2003, 29–34.

Lecture	Papers
Lesson 10: RT and Multimedia	1. Goel, Abeni, Krasic, Snow, Walpole, “Supporting Time-Sensitive Applications on a Commodity OS,” OSDI 2002. 2. Broomhead, Cremean, Ridoux, Veitch, “Virtualize Everything but Time,” OSDI 2010. 3. Hilley & Ramachandran, “Persistent Temporal Streams,” Middleware 2009. 4. Shahabi, Zimmermann, Fu, Yao, “Yima: A Second-Generation Continuous Media Server,” IEEE Computer, June 2002.
Lesson 11: Security	1. Saltzer & Schroeder, “Protection and the Control of Information in Computer Systems,” Proceedings of the IEEE 63(9):1278–1308, Sept 1975. 2. Satyanarayanan, “Integrating Security in Large-Scale Distributed Systems,” TOCS, Aug 1989.

Note on partial readings: Partial readings without specified sections are not strictly required for exams and projects; however, if content is covered in lecture or the hangout, that content is fair game. If you choose a partial reading for a paper summary, you must read and summarize the entire paper.

Grade Distribution

- **Pre-lab:** 2%
- **Homework:** 3%
- **Projects:** 32%
 - Project 1: 8%
 - Project 2: 8%
 - Project 3: 8%
 - Project 4: 8%
- **Class Participation:** 5%
 - Ed discussion: 3%
 - Two paper summaries: 2%
- **Tests:** 58%
 - Test 1: 20%
 - Test 2: 20%
 - Test 3: 18%

Project administration

- **Project code:** All course projects should be completed **INDIVIDUALLY**. AI tools are allowed during implementation, but students must disclose their AI usage and provide the requested AI logs. Regardless of AI tool use, **students are responsible for understanding all code they submit**. We will use an autograder to grade the project for correctness and meeting the stated requirements and generate a raw score (R).

- **Project Interviews:** Students have to sign up for a short 15-minute oral interview. The intent is to verify the student's understanding of their submitted work. The interview score (m) will be between 0 and 1: 0 if the student shows no understanding of the submitted project; 1 if the student shows complete understanding. We will provide a high-level writeup of what you can expect during the project interview.

- **Project grade:** The project grade will be:

$$R \times m$$

- **Interview signup:** Students are responsible for signing up for an interview slot by the announced deadline. Missing the interview without a valid documented reason will result in a zero for the project. Late by more than 5 minutes for the interview OR last-minute rescheduling within 24 hours of the signed-up interview slot will result in a 25% score reduction:

$$\text{final score} = R \times m \times 0.75$$

Class Participation grade details: Provide answers to peer questions; ask questions; collaborate on past exams. We will use forum summary stats ("views," "contributions," "questions," "answers"). Only meaningful contributions will be considered; students should avoid spamming the forum with comments like "thank you", "great job", or "nice" unnecessarily.

Paper summaries: Sign up on the Wiki and choose two papers from the reading list to summarize (≈ 1 page each). If you choose a partial reading to summarize, read and summarize the entire paper.

Sign-up link: [CS6210 OMSCS Fall 26 Paper Summary Sign Up](#)

Exam Administration

- Tests are proctored with **Honorlock**.
- You may use one sheet of **blank** scratch paper. Show both sides to the webcam before starting.
- 80% of questions are released during the exam window. You may collaborate beforehand, but the exam itself is timed, closed-notes, individual, and proctored.

Extra Credit

- **Video Hangout attendance:** 0.5% for at least 10 appearances (attendance via PointSolutions quizzes; quizzes are ungraded and for attendance/comprehension).
- **Alternative (time-zone constraints):** 0.5% via an extra-credit assignment:
 - Summarize any 10 hangout recordings, ≤ 1 page each.
 - Submit as a single PDF.
 - Eligible only if you cannot attend hangouts and mark attendance via Turning Point.
- **CiOS bonus:** If final CIOS completion rate $> 85\%$, everyone receives $+0.5\%$ to course total.

Course Policies

Honor Code

All students must know and follow the Georgia Tech Honor Code.

See: [GT Student Honor Code](#).

Not allowed:

- Copying, with or without modification, someone else's work when this work is not meant to be publicly accessible (e.g., a classmate's program or solution).
- Submission of material that is wholly or substantially identical to that created or published by another person or persons, without adequate credit notations indicating authorship (plagiarism).
 - For project submission, if AI tools are used then the student should document clearly which tools, and what prompts were provided.
- Putting your projects on public Github.
- You are encouraged to discuss problems and papers with others as long as this does not involve copying of code or solutions. Any public material that you use (AI tools, open-source software, help from a textbook, or substantial help from a friend, etc.) should be acknowledged explicitly in anything you submit to us. If you have any doubt about whether something is legal or not please do check with the class Instructor or the TA.

Zero tolerance for any infraction. It pains me to say this but there have been incidents of infractions in previous semesters and the consequence is very serious and will affect your academic standing severely. So please do not put yourself (or me) in that situation.

Collaboration Policy

In order to facilitate a collaborative learning environment, you may discuss the projects and different approaches and techniques, as well as problems you may be having with each other. However, **all the code you turn in must be the result of your individual effort (including prompts to AI tools)**. For written homework, similarly, you may discuss the problems and concepts, but the actual answer that you submit must be written entirely by you. Please note that any infractions will be reported to the Office of Student Integrity (OSI) and could lead to suspension from the OMS program.

Late Policy

- Late submissions accepted for 24 hours after a deadline with a 20% penalty.
- **One Time Forgiveness (OTF):**
 - Each student has **one OTF** during the semester, usable on any graded work **except exams**.
 - An OTF allows submission up to **1 week past the posted deadline with no penalty**.
 - Late submissions are accepted for 24 hours after the OTF deadline (i.e., 8 days after the original deadline) with a 20% penalty.
 - To use OTF, complete the OTF request form here: [CS 6210 OTF Form](#). Please check your spam or junk folder for the confirmation email.
 - If you face any issue, please create a private Ed post.