NYU TANDON SCHOOL OF ENGINEERING

Course Syllabus

Computer Science and Engineering

Introduction to Operating Systems

Course Information

Course Pre-requisites

Graduate student status.

Course Description

This is a graduate-level course that covers an introduction to operating systems. It is geared at graduate students who have not had a good undergraduate course that covers the principles of operating systems. This material is critical for understanding anything about what is really going on in your desktop, your laptop, or in your data center. While it covers material that often finds its place in undergraduate curricula, we will be supplementing that material with some more recent results from the systems literature. In addition, there is the opportunity to customize the material based on overall student background entering the course. I will also cover some recent research on operating system security, virtual machine based security and other related topics in virtual classrooms.

Course Objectives

This course will provide students with the opportunity to:

Understand state-of-the-art in network protocols, architectures, and applications process of networking research - Constraints in thought process of networking research.

Course Structure

This course is conducted entirely online, which means you do not have to be on campus to complete any portion of it. You will participate in the course using NYU Classes located at https://newclasses.nyu.ed
Your final grade will be computed as a combination of the components shown below.

- Midterm exam: 30%
- Assignments: 30%
- Student Research Topic Presentation: 10%
- Final Exam: 30%

Weekly Structure

Week 1: Overview of Computer Systems

- Review of basic relevant computer architecture issues.

Week 2: Evolution of Modern Operating Systems

- History of operating systems
- Operating system basics

Week 3: Processes

- What is a process?
- Process states
- How to characterize a process?
- How the OS Controls a Process?

Week 4: Threads

- Invention of threads
- Applications of threads
- User-level v/s kernel threads

Weeks 5-6: Synchronization primitives and deadlocks

- Concurrency challenges
- Race condition, critical section, mutual exclusion, etc.
- Multiple approaches to enforcing mutual exclusion
- Readers-Writers Problem
- Deadlock, a Deadlock Protection and Avoidance.

Week 7-8: Memory management

- Requirements for Memory Management
- Memory Partitioning (Fixed v/s Dynamic)
- Paging and Segmentation
- Basic Idea of Virtual Memory
- Hardware and Software Support for VM
- Glimpse of Linux Memory Management

Week 9: CPU Scheduling

- Basic objectives for scheduling
- Scheduling Algorithms
- Glimpse of the Linux scheduler

Week 10-11: I/O Management and Disk Scheduling

- Basic issues in I/O, geometry of disks, disk scheduling
- File system functionality
- Different methods of file organization

- Different methods of secondary storage management

Weeks 12-15: Advanced Topics

- File system architectures (Fast file system, Log structured file system)
- Virtualization
- Introduction to distributed systems

Learning Time Rubric

Please modify the below table to represent the breakdown of learning time in each week of your course.

Learning Time Element	Asynchronous* / Synchronous**	Time on Task for Students (weekly)	Notes
Reading Assignments / Recorded Lecture	Asynchronous	2.5 hours	Video format. Expect quizzes throughout the module or weekly chapter readings
Weekly Discussion Board	Asynchronous	1.5 hours	Students are expected to post initial response to weekly topic questions. See Interaction Policy.

Assessment (Labs and Programming assignments)	Asynchronous	2 hours	Students submit their assignment by [the end of the week]
Reading Assignment	Asynchronous	2 hours	Reading assigned textbook chapters and journal articles.
Live webinars	Synchronous	2 hours	Group discussion in class, live, overly weekly chapter

Course Communication

Interaction Policy

Please follow the interaction guidelines stated below for this course.

- I will be holding online virtual classroom sessions every week. This virtual classroom will be held via NYU Classes on Thursdays from 8am to 9am.
- The course will involve regular discussions via the Discussion Forums within NYU Classes and students are encouraged to participate.
- If you have a technical or course content related question, please send me an email. If I think that your question can benefit the class, I might post it on the discussion forum.
- If you have a question related to grading, please send an email to the TA and cc on the email thread. The TA will be responsible for examining your answers and providing a grade as per my guidelines.

- If any other questions need to be answered that are not addressed via email or the live classroom, I can hold virtual office hours on an appointment basis.

Announcements

Announcements will be posted on NYU Classes on a regular basis. You can locate all class announcements under the *Announcements* tab of our class. Be sure to check the class announcements regularly as they will contain important information about class assignments and other class matters.

Email

You are encouraged to post your questions about the course in the Forums discussions on NYU Classes. This is an open forum in which you and your classmates are encouraged to answer each other's questions. But, if you need to contact me directly, please email me. All homework, labs or programming assignments related questions must be researched first on own time, then posted on forums, then discussed with TAs during weekly reviews, and then can be forwarded to me. Typically, you can expect a response within 48 hours.

Readings

The required text for the course is: Andrew S. Tanenbaum and Herbert Bos. 2014. *Modern Operating Systems* (4th. ed.). Prentice Hall Press, USA.

You can access NYU's central library here: http://library.nyu.edu/

You can access NYU Tandon's Bern Dibner Library here: http://library.poly.edu/

RECOMMENDED READINGS are online journal articles provided in each lecture You can access NYU's central library here: http://library.nyu.edu/

You can access NYU Tandon's Bern Dibner Library here: http://library.poly.edu/

Assignments and Exams

Exams Administered and Proctored Online

Exams in this course are administered through NYU Classes. You are required to arrange an online proctor for your exams via ProctorU. More information on ProctorU and scheduling proctoring sessions can be found on Tandon Online's website.

Exams Administered On Paper and Proctored Remotely

Exams in this course are administered via paper and pencil. If you are not able to attend an exam session on-campus, you are required to secure in-person proctoring arrangements near your location. Tandon Online's website.

University Policies

Moses Center Statement of Disability

Academic accommodations are available for students with disabilities. Please contact the Moses Center for Students with Disabilities (212-998-4980 or mosescsd@nyu.edu) for further information. Students who are requesting academic accommodations are advised to reach out to the Moses Center as early as possible in the semester for assistance.

NYU Tandon School of Engineering Policies and Procedures on Academic Misconduct[1]

A. Introduction: The School of Engineering encourages academic excellence in an environment that promotes honesty, integrity, and fairness, and students at the School of Engineering are expected to exhibit those qualities in their academic work. It is through the process of submitting their own work and receiving honest feedback on that work that students may progress academically. Any act of academic dishonesty is seen as an attack upon the School and will not be tolerated. Furthermore, those who breach the School's rules on academic integrity will be sanctioned under this Policy. Students are responsible for familiarizing themselves with the School's Policy on Academic Misconduct.
B. Definition: Academic dishonesty may include misrepresentation, deception, dishonesty, or any act of falsification committed by a student to influence a grade or other academic evaluation. Academic dishonesty also includes intentionally damaging the academic work of others or assisting other students in acts of dishonesty. Common examples of academically dishonest behavior include, but are not limited to, the following:
    a. Cheating: intentionally using or attempting to use unauthorized notes, books, electronic media, or electronic communications in an exam; talking with fellow students or looking at another person's work during an exam; submitting work prepared in advance for an in-class examination; having someone take an exam for you or taking an exam for someone else; violating other rules governing the administration of examinations.
    b. Fabrication: including but not limited to, falsifying experimental data and/or citations.
    c. Plagiarism: intentionally or knowingly representing the words or ideas of another as one's own in any academic exercise; failure to attribute direct quotations, paraphrases, or borrowed facts or information.
    d. Unauthorized collaboration: working together on work that was meant to be done individually.
    e. Duplicating work: presenting for grading the same work for more than one project or in more than one class, unless express and prior permission has been received from the course instructor(s) or research adviser involved.

[1] Excerpted from the Tandon School of Engineering Student Code of Conduct

f. Forgery: altering any academic document, including, but not limited to, academic records, admissions materials, or medical excuses.