



CSE 421/521: Operating Systems

TTh 11:00-12:20 pm

Norton 112



Karthik Dantu
kdantu@buffalo.edu
<http://www.cse.buffalo.edu/faculty/kdantu>
Davis 106
Office Hours: TBD
(716) 645-2670

Course Description: This course introduces the design and implementation of Operating Systems. It will focus on different OS design techniques, process management, processor scheduling, concurrent programming, deadlocks and synchronization, memory management, file management and I/O systems, disk scheduling, protection and security, and distributed systems. The course will include hands-on programming projects on different OS components.

Prerequisite(s): CSE 250, or an equivalent course, or permission of the instructor.

Credit Hours: 4 (421)/ 3 (521)

Text(s): None. This course does not require any textbooks. The only required material will be class lectures and slides.

Recommended Book: *Operating Systems: Principles and Practice*

Author(s): Thomas Anderson and Mike Dahlin

ISBN-13: 978-0985673512

Supplementary text: *Operating Systems Concepts*

Author(s): Abraham Silberschatz

ISBN-13: 978-1118063330

Course Objectives:

At the completion of this course, students will be able to describe basic functionality of an Operating System, internal working of threads and processes, various scheduling techniques for CPU scheduling, deadlocks and deadlock avoidance, memory management in a multi-tasking environment, virtual memory, design of file systems, and some of these concepts extended to distributed systems.

Grade Distribution:

Programming Assignments in PintOS	45% (3 PAs of 15% each)
Midterm Exams	40% (Two midterms of 20% each)
In-class quizzes	10%
Class and online participation	5%

Grades will be on a curve. From prior experience, I have seen different performance from the undergraduate (421) and graduate (521) students. Therefore, for fairness, I'll grade students on two separate curves, one for graduate students and another for the undergraduate students.

Course Policies:

- **General**

- Computers are not to be used unless instructed to do so.
- Exams are closed book, closed notes.
- **No makeup quizzes or exams will be given unless discussed on a per case basis well in advance of the exam (at least two weeks).**

- **Grades**

- Graded on a curve with B+ roughly being the median.
- Grades will be maintained in the myUB course system. Students are responsible for tracking their progress by referring to the online gradebook and report any discrepancies.

- **Labs and Assignments**

- Students are expected to work independently. **Offering** and **accepting** solutions from others is an act of **plagiarism**, which is a serious offense and **all involved parties** will be penalized according to the **Academic Honesty Policy**. Discussion amongst students is encouraged, but when in doubt, direct your questions to the professor, or teaching assistant.
- Unless addressed with the professor well ahead of a deadline, late submissions will be penalized. 10% penalty for submissions that are 0-24 hrs late; 25% penalty for submissions that are 24-48hr late; 50% for submissions that are 48-72 hr late and 100% after.

- **Attendance and Absences**

- Attendance is expected each class. Participation in class and on piazza carries 5% of the grade. Additionally, there will be in-class quizzes that will account for 10% of the grade. The professor will not entertain any grade changes toward the end of the course if the student has not participated during the semester.
- Students are responsible for all missed work, regardless of the reason for absence. It is also the absentee's responsibility to get all missing notes or materials.

Plagiarism Policy:

- This course will have three programming assignments based on the PintOS programming environment. We use **sophisticated code checkers** to check for code copied from assignments from this class as well as submissions from prior editions of the class. It is **very improbable** that you will be able to fool the code checker.
- Along the same lines, if you see someone else's code to understand the logic, it is probable that our code checker will flag this as plagiarized since your code will be influenced by what you saw and will look structurally similar.
- Discussing programming logic with other students is acceptable. However, this is a slippery slope, and the more detailed these discussions are and the more they tend toward actual code, the more likely the code checker will flag this as plagiarism.
- Copying snippets of code from online resources is also considered plagiarism. When you are in doubt, please check with the professor for clarity.