

## Course Description

An Operating System is a complex software package that manages the resources of a computer system, and provides the base upon which applications can be written. In this course we will study the basic components of an operating system, their functions, mechanisms, policies and techniques used in their implementation and several examples from popular operating systems. The components, which will be discussed, include:

- Process management: process description and control, system calls, concurrency, mutual exclusion, synchronization, inter-process communication, deadlock and scheduling.
- Multiprogramming and concurrency using processes. Programming with threads: creation, multi-threaded programs, synchronization, and scheduling.
- Storage management: virtual memory, I/O management and file systems.
- Networking and distributed systems: network protocols, two-tier and three-tier client/server application development; issues in building a distributed systems.
- Protection and security: protecting resources, security threats, public key encryption, access control, and network security.

Hands on practical projects in C++ will support concepts discussed during the lecture. On completion of this course students will be able to understand the components and working of an operating system and the design and implement various operating system functions.

## Course Information

Newsgroup:	sunyab.cse.421
Website:	<a href="http://www.cse.buffalo.edu/~bina/cse421/fall2006">http://www.cse.buffalo.edu/~bina/cse421/fall2006</a>
Instructor:	Bina Ramamurthy (bina@cse.buffalo.edu)
Lecture Time:	MWF: 9.00-9.50AM
Lecture Location:	114 Hochstetter
Office:	127 Bell Hall
Office Hours:	MW 10.00-11.30AM
Recitation A1:	T, 11.00 – 11.50 AM, 115 Talbert
Recitation A2:	W 10.00 – 10.50 PM, 214 Norton
Recitation A3:	F 3:00 – 3:50 PM, 209 Norton

## Textbook and other material

The primary textbook for this course is:

**Modern Operating System by Andrew S. Tannenbaum, Second Edition, Prentice-Hall Inc., 2001.**

While there are no other required textbooks, you should have in your possession appropriate reference books for both the C and C++ programming languages. I would strongly suggest Kernighan and Ritchie's **The C Programming Language** and Stroustrup's **The C++ Programming Language**.

## Pre-requisites

CSE421 is the capstone/core course for your degree within the Department of Computer Science and Engineering. The course requires the use of many skills that you have developed and refined over the last several semesters. Due to the skill level required and inherent difficulty of this course, it is required that you have **successfully completed** the pre-requisite courses. **Successful completion** means that you have completed the course in a semester **prior** to the current one, and that you achieved a grade of C- or higher. The pre-requisites for this course include **CSE241/EE378 and CSE305 or equivalent courses**. If you do not possess the

pre-requisites, you must drop the course immediately. Failure to do so will result in the department dropping you from the course *at your expense*. Computer Architecture will help with the hardware concepts, and 305 will provide broader exposure to programming languages.

You will also be working on several large programming projects over the course of the semester. You must have a strong working knowledge of C (intermediate level background or above). You must also have enough experience with C++ to understand the fundamentals of classes. This knowledge should extend to dynamic allocation and de-allocation of instances and fundamental pointer operations for class instances.

**NOTE:** It is **NOT** the intent of this course to teach you how to program in a new programming language. I assume that since you are all seniors in Computer Science and Engineering you should be able to pick up the essentials of any programming language within a few weeks. Much of the lab work revolves around strong design, which you have all been exposed to since Programming/Computer Science I.

## Grading Distribution

Grades will consist of the following components:

Component (Quantity)	Percentage
Labs (3)	15%, 20%, 20%
Midterm (1)	20%
Final (1)	25%

Point distribution guideline will be as follows:

Point Range	Letter Grade
95.00-100	A
90.00-94.99	A-
85.00-89.99	B+
80.00-84.99	B
75.00-79.99	B-
70.00-74.99	C+
65.00-69.99	C
60.00-64.99	C-
55.00-59.99	D+
50.00-54.99	D
0-49.99	F

I reserve the right to alter component weighting or provide a “curve” on an assignment as warranted.

**NOTE :** → **COMPONENT PASS POLICY** ← IN ORDER TO PASS THIS COURSE, YOU MUST HAVE PASSING WEIGHTED COMPONENT AVERAGES (WEIGHTED COMPONENT AVERAGES MUST BE GREATER THAN 49.99) THERE WILL BE TWO COMPONENTS THIS SEMESTER. COMPONENT 1 IS THE EXAM COMPONENT CONSISTING OF THE MIDTERM AND FINAL. COMPONENT 2 IS THE LAB COMPONENT CONSISTING OF THE THREE PROJECTS.

## Labs

Lab assignments constitute a major portion of the course. Over the semester, you will be given three lab assignments. The lab experiments will cover fundamental areas of Operating System development. You will be given approximately 25 – 30 days to complete each exercise. *Do not be lulled into a safe sense of security. Do not think you have a lot of time to implement each lab!*

All due dates will be clearly printed on the top of each assignment. Late assignments will not be accepted **for any reason**. *Remember, it is better to submit your solution every so often before the due date!*

Develop your code using the *Incremental Development technique*. Do not try to sit down and code the entire assignment in one sitting. Instead, take one section at a time, implement, test it, back up the code, and move on to the next section. You will turn in each lab before 11:59 PM on the due date via the departmental *submit* command. You must also include appropriate testing programs to show the validity of your solution. In addition, you must include external documentation discussing the “how’s and why’s” of your design and implementation. You will be required to demonstrate your lab to your TA. The TA will also run test examples against your code to check your solution’s overall correctness. The TA will provide a demo schedule. It is your responsibility to demo your project, or you will receive a zero for that portion of the grade.

When your grade is assigned for the lab, the TA will indicate critical areas that must be fixed in order to solve the next assignment. In general, solutions are not provided for the lab. It is not that we do not have them, or am unwilling to distribute them, it’s that there is no one answer to any project solution.

## Exams

There will be a Midterm that will be administered and graded before the resign date. Midterm material will cover all lecture and reading assignments before the exam, as well as concepts from the lab assignments. Midterms are closed book, closed notes, and closed neighbor. Please see the additional handout for exam taking policies for this course.

The final is a comprehensive exam, covering all lecture, lab, and homework areas. The final is closed book, closed notes, and closed neighbor. Please see the additional handout for exam taking policies for this course.

We do not give make up exams for any reason. If you miss an exam, you will receive a zero for that portion of the grade.

## Attendance Policy

You are responsible for the contents of all lectures and recitations (your assigned section). If you know that you are going to miss a lecture or a recitation, have a reliable friend take notes for you. Of course, there is no excuse for missing due dates or exam days. We do, however, reserve the right to take attendance in both lecture and recitation. We may use this information to determine how to resolve borderline grades at the end of the course, especially if we see a lack of attendance and participation during lecture sessions.

During lectures, we will be covering material from the textbook. We will also work out several of the problems from the text. Lecture will also consist of the exploration of several real world Operating System problems not covered in the book. You will be given a reading assignment at the end of each lecture for the next class.

Recitations are designed to review difficult concepts in the class and to spend additional time discussing the lab work required for the course. The recitation is your time to communicate with your TA about the course. Use the opportunity to the fullest.

## Office Hour Policy

If you can’t meet during these hours, you will have to communicate with us via Email. Office hours are intended to resolve questions about the material that could not be answered in lecture or recitation. Come to office hours prepared! Office hours are **NOT** for the following: to repeat missed lecture material, to repeat missed recitation material or to have the instructor or TA solve an assigned problem for you. During office hours, we will **NOT** write or debug your code for you! Instead, we will direct you as to where to concentrate your debugging efforts.

## Grading Policy

All assignments will be graded and returned in a timely manner. When an assignment is returned, you will have a period of one week to contest any portion of the grade. The TA who graded your assignment will be the first person to resolve a grading conflict. If the conflict cannot be resolved, the instructor will mediate the dispute. The judgment of the instructor will be final in all such cases. When contesting a grade, you must be able to demonstrate how your particular solution is correct. Also, when contesting a grade, the instructor or TA reserves the right to re-evaluate the entire lab or exam, not just the portion in dispute.

## Incomplete Policy

We only grant incompletes in this course under the direst of circumstances. By definition, an incomplete is warranted if the student is capable of completing the course satisfactorily, but some traumatic event has interfered with their capability to finish within the timeframe of the semester. Incompletes are not designed as stalling tactic to defer a poor performance in a class.

## Academic Integrity Policy

UB's definition of Academic Integrity in part is, "Students are responsible for the honest completion and representation of their work". It is required as part of this course that you read and understand the departmental academic integrity policy located at the following URL:

[http://www.cse.buffalo.edu/academics-academic\\_integrity.shtml](http://www.cse.buffalo.edu/academics-academic_integrity.shtml)

There is a very fine line separating conversation pertaining to concepts and academic dishonesty. You are allowed to converse about general concepts, but in no way are you allowed to share code or have one person do the work for others. You must abide by the UB and Departmental Academic Integrity policy at all times. **NOTE:** Remember that items taken from the Internet are also covered by the academic integrity policy! If you are unsure if a particular action violates the academic integrity policy, assume that it does until you receive clarification from the instructor.

*This semester, all projects will be checked using an electronic cheat checking system. We reserve the right to check or question any portion of any work submitted at any time during the semester or afterwards. If you are caught violating the academic integrity policy, you will minimally receive a ZERO in the course.*

## Web Site and Newsgroup

The CSE421 website should be checked frequently for important news. Course assignments, slides, grade reporting, and general hints and tips will be posted on the website. The newsgroup will be used as a method to ask questions regarding assignments. We encourage all students to participate in newsgroup conversations. Generally, we will try to answer newsgroup questions within a period of 48 hours of their posting.

## Students with Disabilities

If you have special needs due to a disability, you must be registered with the Office of Disability Services. If you are registered with ODS please let your instructors know about this so that they can make special arrangements for you.