CSE 4/521 Introduction to Operating Systems Spring 2005

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++) using experimental operating system Nachos 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 to design and implement various operating system functions.

Course Information

Newsgroup:	sunyab.cse.421
Website:	http://www.cse.buffalo.edu/~bina/cse421/spring2005
Instructor:	Bina Ramamurthy (bina@cse.buffalo.edu)
Lecture Time:	CSE421/521: MWF: 12:00-12:50PM
Lecture Location:	228 NSC
Office:	127 Bell Hall
Office Hours:	MW: 1.00-2.20PM
Recitation B1:	Mon 11.00 – 11.50AM, Clemens 19, TA:
Recitation B2:	Tue 4.00 – 4.50PM, Baldy 113, TA:
Recitation B3:	Wed 8.00 – 8.50AM, Clemens 209, TA:

Textbook and other material

The primary textbook for this course is:

Operating System Concepts by Siberchatz, Galvin and Gagne, Seventh Edition, John-Wiley and Sons, 2005.

While there are no other required textbooks, you should have in your possession appropriate reference books for both the C and C++ programming languages.

Pre-requisites

CSE421 is *the* capstone course for your Bachelor 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 CSE250 or an equivalent course.*** 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.***

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. I assume that since you are all seniors and graduates in Computer Science and Engineering you should be able to pick up the essentials of any programming language within a few weeks.

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 assignments will require you to alter and extend the capabilities of an instructional Operating System called ***Nachos***. The Nachos Operating System provides both the instructor and the student with the ability to explore important implementation concepts without the hassle of a stand-alone development machine. 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!* Although many parts of the Nachos lab assignment only require a few lines of code to implement, each lab does require that you have a strong understanding of the existing code. This understanding takes time, patience, and an experimenting attitude.

Remember, it is better to submit your solution every so often before the due date!

For each project, we will have a Nachos Walk-Through day in lecture. During these walkthroughs, we will show you what areas of code to look at, what you need to do, and different approaches to help you design your solution. During walk-through days, we may constrain or relax some of the conditions of the lab assignments, so it is important to attend!

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. During testing, you will compile and run your Nachos code for the 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.

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. 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. I 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
Although attendance is not officially required, you are still 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. A Detailed lecture schedule is enclosed. 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/undergrad/academicintegrity.html.

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 me. *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. We will also place the incident in your permanent record. If it is your second violation, we will recommend to the Director of Undergraduate Studies that formal proceedings be filed against you, which would mean either you could be expelled, or your degree progress will be terminated within the Computer Science and Engineering department. ***WE TAKE ACADEMIC HONESTY QUITE SERIOUSLY, SO SHOULD YOU!***

Web Site and Newsgroup

The course 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 me know as soon as possible.

Tentative Lecture Schedule:

Date (Week of)	Topics Covered	Reading Material
1/17	General Introduction; Course Outline; Computer hardware review; Operating Systems (OS) fundamentals. Different OS structures.	Ch.1, Ch.2
1/24	Introduction to Nachos operating systems. Project 1 discussion: System calls and Exception handling	Project 1 handout
1/31	Different OS structures. Process model; Process description and control; examples from Nachos.	Ch.3
2/7	The Thread Model; Implementing concurrency; multithreaded programs; Nachos threads.	Ch.4, Nachos
2/14	Process communication and synchronization: semaphores, monitors, messages, and locks.	Ch. 6
2/21	Classical IPC Problems and solutions. Exam Review. Exam 1: 2/25.	Ch.6; Exam on Ch.1- 4 and 6.
2/28	Project 2 discussion	Project 2 handout
3/7	CPU scheduling: long term, short term and real-time scheduling. 3/14: Spring Break	Ch. 5
3/21	Deadlock detection and resolution; avoidance and prevention.	Ch.7
3/28	Memory management. Virtual memory and demand paging. Project 3 discussion	Ch. 8 and Ch.9; Project 3 handout
4/4	File system design and implementation; IO subsystem and mass storage	Ch. 10, Ch.11, and Ch.12
4/11	Security and protection; authorization and authentication; Trusted systems.	Ch. 14 and 15
4/18	Sample operating systems: Windows XP, .net, and Linux	Ch.19 and Ch. 21
4/25	Review for the final exam. Final exam during final exam week.	
Important Dates	**Due Date**	
2/27	Project 1	
4/3	Project 2	
5/1	Project 3	
2/25	Exam 1	
Finals Week	Final Exam	