ECE 329 Computer Systems Structures

Fall 2007

Instructor:	Stan Birchfield
Office:	207-A Riggs Hall, 656-5912, stb at clemson
Office Hours:	4:00-5:00 M-F, or by appointment
Course website:	**http://www.ces.clemson.edu/~stb/ece329**
Text:	A. S. Tanenbaum, *Modern Operating Systems*, 2nd edition, 2001
Prerequisites:	ECE 222; ECE 272.

Overview: In this course students will learn the basics of operating systems, including the creation, management, and scheduling of threads and processes; process communication and synchronization; deadlocks; memory management; file systems; I/O; and protection. Programming assignments provide a practical connection to the material.

Objectives: By the end of the course, students should be able to do the following:

- *Fundamental concepts.* Define, explain, and use the basic concepts and terms of operating systems and structures, such as threads, processes, starvation, thrashing, spinlock, deadlock, critical regions, atomic transactions, mutexes, semaphores, process states, graceful degradation, frames, fault-tolerant systems, remote procedure calls, inodes, shared memory, and message passing. Explain the difference between user and kernel mode, hard and soft real-time systems, logical and physical addresses, segmentation and paging, internal and external fragmentation, I/O-bound and CPU-bound process, preemptive and non-preemptive scheduling.
- *Specific computer systems.* Compare and contrast the design choices of modern operating systems, including Windows and Linux/Unix. Explain the details of implementation with regard to the Intel IA-32 architecture.
- *Programming skills.* Write clean, well-documented multithreaded C/C++ code utilizing thread creation, thread suspension, thread cancellation, mutexes, and semaphores.

Topics:	**Lectures**
• introduction	1
• C/C++ programming language and tools	3
• processes and threads	4
• process synchronization	4
• interprocess communication	3
• CPU scheduling	4
• deadlocks	3
• memory management	4
• I/O systems	3
• file systems	4
• multi-processor systems	3
• Windows/Linux/Unix and IA-32	6
• tests	1

TOTAL	43

Grading:

1. *Tests.* There will be one midterm and a final examination, both of which will be closed-book, closed-notes. The instructor should be notified at least one week in advance of any conflict that would prevent a student from taking an exam, so that alternative arrangements can be made. Without prior approval, missed tests cannot be made up except in cases of extreme urgency and importance (e.g., sudden illness, death in the immediate family).
2. *Quizzes.* There will be several short, unannounced quizzes during class (approximately one per week). A student who keeps up with lectures and reading assignments should do well on these quizzes. Unless agreed upon in advance by the instructor, missed quizzes will receive a grade of zero. To receive a passing grade in the course, a student must miss no more than three quizzes. The grade of the lowest quiz may be dropped, subject to instructor discretion.
3. *Programming Assignments.* There will be six or seven programming assignments. Late assignments will be accepted at a penalty of 10 points per day (according to a six-day work week), up to a maximum of 30 penalty points; assignments turned in more than one week late will receive a grade of zero. Although students are encouraged to discuss the assignments with their colleagues and to search the web to aid their understanding of the material, they must turn in their own work. Copying the code of others is strictly prohibited. Students who violate University rules on academic dishonesty will be subject to disciplinary penalties, such as failure in the course and/or dismissal from the University.
4. *Grading.* Grades will be determined by the following formula: midterm (30%), final exam (30%), programming assignments (30%), quizzes (10%).