ECE 329 Computer Systems Structures
Fall 2005

Instructor: Stan Birchfield, 207A Riggs Hall, 656-5912, stb at clemson
Office Hours: 9:00-10:00 MWF, or by appointment

Grading assistant: Nikhil Rane, nrane at clemson

Class meets: 8:00-8:50 MWF, 227 Riggs Hall

Website: **http://www.ces.clemson.edu/~stb/ece329**

Text:

- A. Silberschatz, P. B. Galvin, and G. Gagne, *Operating System Concepts*, Sixth Edition, John Wiley & Sons, 2003.

Prerequisites: CPSC 102 or 210; CPSC 340 or 212; 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; memory management; file systems; and protection. Programming assignments connect the theory with practice and enable students to further develop their programming skills.

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

- *Fundamental concepts.* Define and explain the basic concepts and terms of operating systems and structures, such as thread, process, starvation, thrashing, spinlock, deadlock, critical region, atomic transaction, mutex, semaphore, the five process states, graceful degradation, frame, fault-tolerant system, remote procedure call, and file table. Explain the difference between 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. Contrast the different types of message passing. Compute the effects of different scheduling algorithms, and apply the Banker's Algorithm.
- *Programming skills.* Write clean, well-documented C and C++ code to achieve desired functionality, along with a clear report. Implement basic operating system functionality on Unix/Linux using fork, signal, alarm, setjmp, longjmp, sigsetjmp, and siglongjmp. Write multithreaded code on Windows using CreateThread, CreateMutex, CreateSemaphore, and WaitForSingleObject.

Grading: midterms (20% each), final (30%), programming assignments (30%)

Topics:	**Lectures**
• introduction	1
• C/C++ programming language	6
• computer and operating system structures	2
• processes and threads	5
• CPU scheduling	5
• process synchronization	6
• deadlocks	5
• memory	3
• file systems	2
• I/O systems	2
• projects	5
• tests	1

	43

Academic integrity: "As members of the Clemson University community, we have inherited Thomas Green Clemson's vision of this institution as a 'high seminary of learning.' Fundamental to this vision is a mutual commitment to truthfulness, honor, and responsibility, without which we cannot earn the trust and respect of others. Furthermore, we recognize that academic dishonesty detracts from the value of a Clemson degree. Therefore, we shall not tolerate lying, cheating, or stealing in any form." – from the university's web page on academic integrity: http://www.cs.clemson.edu/html/academics/academic_integrity_2002.html