ICS 431 Syllabi

Designer:	Dr. Farag A Azzedin
Date:	30/03/2005
Course Number and Title:	ICS431: Operating Systems (3-3-4)
Course Description:	This course introduces the fundamentals of operating systems design and implementation. Topics include history and evolution of operating systems; Types of operating systems; Operating system structures; Process management: processes, threads, CPU scheduling, process synchronization; Memory management and virtual memory; File systems; I/O systems; Security and protection; Case operating systems.
Course Objectives:	1. Introduce fundamental concepts and principles of operating systems. 2. Expose students to popular operating systems such as Unix/Linux and Windows. 3. Provide hands-on experience through operating systems programming projects carried out in the lab.
Course Learning Outcomes:	Upon the completion of this course, students should be able to: **<u>For course objective 1</u>** 1. Identify the role of operating systems. [EM: 1 – 4] 2. Explain the different structures of operating systems. [EM: 1 – 4] 3. Identify the differences between mode switching and process switching. [EM: 1 - 4] 4. Describe OS support for processes/threads, and virtual memory. [EM: 1 – 4]Evaluate processes and/or threads synchronization mechanisms. [EM: 1 – 5] 5. Explain deadlock conditions and ways to resolve them 6. Recognize the tradeoffs in applying different algorithms for CPU scheduling and disk scheduling. [EM: 1 – 5] 7. Describe OS support for I/O and file systems. 8. Explain protection and security issues related to operating systems. [EM: 1 – 5] **<u>For course objective 2</u>** 1. Identify the different design and implementation concepts for Unix/Linux and Windows. [EM: 1 – 5] 2. Evaluate and select the proper operating system for a particular computing environment

	<u>For course objective 3</u> 1. Create and manage processes/threads under the Unix/Linux. [EM: 5] 2. Use Inter-Process Communication techniques under Unix/Linux. [EM: 5] 3. Implement synchronization techniques between different processes and/or threads. [EM: 5] 4. Demonstrate the ability to modify and build components of an operating system (such as different CPU scheduling algorithms, different memory management schemes). [EM: 5]
Course Prerequisite(s):	COE 205
Course Prerequisite(s) by Topic	PF2. Algorithms and problem-solving. PF3. Fundamentals data structure. AR3. Assembly level machine organization. AR4. Memory system organization and architecture
Textbook(s) and References:	Approved textbook: 1. *Operating System Concepts* by Siblerschatz and Galvin. 6th Ed. 2001 Addison Wesley, Inc. References: 1. *Modern Operating Systems* by Andrew S. Tanenbaum; 2nd Ed, 2000, Prentice Hall, Inc. 2. *Operating Systems* by William Stallings. 3rd Ed. 1998. Prentice-Hall, Inc.
Evaluation Methods (EM)	A recommended evaluation method: EM1: Homework (4) 0% EM2: Quizzes (4) 10% EM3: Major I 15% Major II 20% EM4: Final Exam 30% EM5: Lab 25%
Weekly Breakdown	1. Introduction (2 hours). 2. Operating System Structure (3 hours). 3. Processes (3 hours). 4. Threads (3 hours). 5. CPU Scheduling (3 hours). 6. Process Synchronization (4 hours). 7. Deadlocks (4 hours). 8. Memory Management (2 hours). 9. Virtual Memory (2 hours).

	10. File System Interface and Implementation (2 hours). 11. I/O Systems and Mass Storage Structure (5 hours). 12. Security and Protection (3 hours). 13. Case Studies (3 hours).
Lab Objectives and Assignments; and Computer Requirements:	1. Illustrate, explore, and implement functions of modern operating systems, including threads and concurrency, multiprogramming, system calls, virtual memory, and file systems. 2. Build some components of an operating system (such as different CPU scheduling algorithms, different memory management schemes) 3. Gain hands-on experience on popular operating systems such as Linux/Unix.
Content Topical Breakdown	OS1 (2 core hours of 2) - Overview of operating systems Modules #1, #2, #13 (& Lab Proj.) OS2 (2 core hours of 2) - Operating System Principles Modules #1, #2 OS3 (6 core hours of 6) - Concurrency Modules #3, #4, #6, #7 (& Lab Proj.) OS4 (3 core hours of 3) - Scheduling and Dispatch Modules #3, #4, #5 (& Lab Proj.) AR4 (3 core hours of 5) – Memory system organization and architecture Module #4, #8, #9 (& Lab Proj.) OS5 (5 core hours of 5) – Memory Management Modules #4, #8, #9 (& Lab Proj.) OS6 (3 hours: electives) – Device Management Module #11 OS7 (2 hours: electives) – Security and Protection Module #12 OS8 (2 hours: electives) -

	File Systems Modules #10 (& Lab Proj.) OS11 (2 hours: electives) - System performance evaluation Modules #5, #11 (& Lab Proj.)
Additional Notes	1. It is highly recommended to use C language in the lab. 2. Exercises and/or projects should be performed using Linux platform. 3. It is recommended that students are introduced to hands-on experience assignments/projects using the Nachos System in the lab. Nachos software, which is an instructional software for teaching undergraduate level operating systems course, is freely available for Linux machines.