

CS450 - Introduction to Operating Systems

Last Updated - 04/02/02

Course Manager – Dr. Xian-He Sun, Associate Professor

3 credit hours; required for CS & CPE; 150 min. lecture each week

Current Catalog Description - Introduction to operating system concepts—including system organization for uniprocessors and multiprocessors, scheduling algorithms, process management, deadlocks, paging and segmentation, files and protection, and process coordination and communication. Prerequisites: (CS 331 and CS 350) or (CS 331 and ECE 242) or (CS 401 and CS 402) or CS 403. (3-0-3)

Textbook

- Siblingschatz and Galvin, *Operating System Concepts*, 6th Ed. 2001 Addison Wesley, Inc..

References - other textbooks or materials

- William Stallings, *Operating Systems*, 3rd Ed. 1998. Prentice-Hall, Inc
- Gray, *Interprocess Communications in UNIX*, 2nd Ed. 1998 Prentice-Hall, Inc.
- Some handouts regarding Unix systems programming will be provided on the course website

Course Goals - Students should be able to:

- Explain the range of requirements that a modern operating system has to address.
- Define the functionality that a modern operating system must deliver to meet a particular need.
- Articulate design tradeoffs inherent in operating system design.
- Explain the concept of a logical layer.
- From the perspective of building operating systems, explain the benefits of building these layers in a hierarchical fashion.
- Describe how the resources of the computer system are managed by software.
- Relate system state to user protection.
- Justify the presence of concurrency within the framework of an operating system.
- Demonstrate the potential run-time problems arising from the concurrent operation of many (possibly a dynamic number of) tasks.
- Summarize the range of mechanisms (at an operating system level) that can be employed to realize concurrent systems and be able to describe the benefits of each.
- Explain the different states that a task may pass through and the data structures needed to support the management of many tasks.
- Compare and contrast the common algorithms used for both preemptive and non-preemptive scheduling of tasks in operating systems.
- Describe relationships between scheduling algorithms and application domains.
- Investigate the wider applicability of scheduling in such contexts as disk I/O, networking scheduling, and project scheduling.
- Introduce memory hierarchy and cost-performance tradeoffs.
- Explain what virtual memory is and how it is realized in hardware and software.
- Examine the wider applicability and relevance of the concepts of virtual entity and of caching.
- Evaluate the trade-offs in terms of memory size (main memory, cache memory, auxiliary memory) and processor speed.
- Defend the different ways of allocating memory to tasks on the basis of the relative merits of each.
- Summarize the features of an operating system used to provide protection and security, and describe the limitations of each of these.
- Summarize the full range of considerations that support file systems.

Prerequisites by Topic

- Data Structure and Algorithms
- Computer Organization and Assembly Language
- Strong Programming Skills

Major Topics Covered in Course

1. Computer system overview	2 hours
2. Operating system overview	2 hours
3. Process description and control	3 hours
4. Threads	3 hours
5. Concurrency: mutual exclusion and synchronization using semaphores and monitors	3 hours
6. Concurrency: deadlock and starvation	6 hours
7. Interprocess Communication: Pipes, Message Queues, Semaphores and Shared Memory	6 hours
8. Memory	5 hours
9. Virtual memory	3 hours
10. Uniprocessor scheduling	3 hours
11. I/O management and disk scheduling	3 hours
12. File management	3 hours
13. Distributed System and Security	3 hours
Midterm Exam	-
Final Exam	45 hours

Laboratory projects (specify number of weeks on each)

- Develop a set of libraries that allow for inter-process message based communication using UNIX message queues, semaphores and shared memory, and pipes. Perform basic latency benchmarking and critique the results. (4 weeks)

Estimate CSAB Category Content in Credit Hours

	CORE	ADVANCED		CORE	ADVANCED
Data Structures	0		Computer Organization and Architecture		1
Algorithms		1	Concepts of Programming Languages	0	
Software Design		1			

Oral and Written Communications - Every student is required to submit at least 0 written reports (not including exams, tests, quizzes, or commented programs) of typically 0 pages and to make 0 oral presentations of typically 0 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

- none

Social and Ethical Issues - Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)?

- none

Theoretical Foundations - Please list the types of theoretical material covered, and estimate the time devoted to such coverage in contact (lecture and lab) hours.

- The majority of the course topics are theoretical and approximately half the lecture time is spent on theory, half on practice. See Major Topics Covered in Course. (20 hours)

Problem Analysis - Please describe the problem analysis experiences common to all course sections.

- Develop a set of libraries that allow for inter-process message based communication using UNIX message queues, semaphores and shared memory, and pipes. Perform basic latency benchmarking and critique the results.

Solution Design - Please describe the design experiences common to all course sections.

- Develop a set of libraries that allow for inter-process message based communication using UNIX message queues, semaphores and shared memory, and pipes. Perform basic latency benchmarking and critique the results.

Other Course Information

- Additional Suggested Course Assignments
 - Assignments will be a mixture of theory exercises and programming problems (C programs in Unix).
- Planned Course Enhancements
 - Include more O/S Programming Projects (see eliminated course CS451 for examples)