CSE 6431 (Approved): Advanced Operating Systems

Course Description
Advanced topics in operating systems and concurrency; introduction to distributed systems.

Prior Course Number: CSE 760
Transcript Abbreviation: Advanced Op Systs
Grading Plan: Letter Grade
Course Deliveries: Classroom
Course Levels: Graduate
Student Ranks: Senior, Masters, Doctoral
Course Offerings: Autumn, Spring
Flex Scheduled Course: Never
Course Frequency: Every Year
Course Length: 14 Week
Credits: 3.0
Repeatable: No
Time Distribution: 3.0 hr Lec
Expected out-of-class hours per week: 6.0
Graded Component: Lecture
Credit by Examination: No
Admission Condition: No
Off Campus: Never
Campus Locations: Columbus
Prerequisites and Co-requisites: CSE 2431 or CSE 5431 or CSE 660
Exclusions: Not open to students with credit for CSE 760
Cross-Listings:

The course is required for this unit's degrees, majors, and/or minors: No
The course is a GEC: No
The course is an elective (for this or other units) or is a service course for other units: Yes

Subject/CIP Code: 14.0901
Subsidy Level: Doctoral Course

Programs

Abbreviation	Description
BS CSE	BS Computer Science and Engineering
MS CSE	MS Computer Science and Engineering
PhD CSE	PhD Computer Science and Engineering

Course Goals

Master language and operating system constructs such as semaphores and monitors and their use for process synchronization and mutual exclusion.
Master principles and algorithms for distributed mutual exclusion in distributed systems.
Be competent with basic issues in developing distributed systems, use of logical clocks, and techniques for interprocess communication.
Be competent with principles and approaches for deadlock detection and avoidance, and fault-tolerance and avoidance in distributed systems.

Be competent with operating system support for database transactions, including ACID properties and serializability of transactions.
Be familiar with modern techniques for fault-detection and failure handling in servers, HPC systems, and data-intensive systems.
Be familiar with use of virtualization and cloud technologies.
Be familiar with software transactional memory (STM) based approaches.
Be exposed to the basic concepts of data consistency and data consistency models.

Course Topics

Topic	Lec	Rec	Lab	Cli	IS	Sem	FE	Wor
Introduction to distributed systems	3.0							
Mutual exclusion and synchronization methods and examples	6.0							
Foundational issues for distributed systems, like clock synchronization and logical clocks	3.0							
Distributed mutual exclusion algorithms	4.5							
Support for database transactions	4.5							
Deadlock detection and avoidance analysis and techniques	6.0							
Advanced Distributed Algorithms and Fault-tolerance methods	3.0							
Modern fault-tolerance and failure-recovery: servers, HPC systems, and data-intensive systems	3.0							
Virtualization technologies and clouds	3.0							
Software Transactional Memory	3.0							
Coherence and Consistency Models	3.0							

Representative Assignments

Use of sychronization methods (semaphores or Java-based support)
Use of Virtualization and Cloud technologies

Grades

Aspect	Percent
Written Assignments	15%
Programming Labs	15%
Mid-Term	25%
Final	45%

Representative Textbooks and Other Course Materials

Title	Author
Distributed Systems	A. Tanenbaum, M. van Steen
Advanced Concepts in Operating Systems	M. Singhal, N. Shivaratri

ABET-EAC Criterion 3 Outcomes

Course Contribution		College Outcome
***	a	An ability to apply knowledge of mathematics, science, and engineering.

Course Contribution		College Outcome
**	b	An ability to design and conduct experiments, as well as to analyze and interpret data.
*	c	An ability to design a system, component, or process to meet desired needs.
	d	An ability to function on multi-disciplinary teams.
**	e	An ability to identify, formulate, and solve engineering problems.
	f	An understanding of professional and ethical responsibility.
	g	An ability to communicate effectively.
	h	The broad education necessary to understand the impact of engineering solutions in a global and societal context.
	i	A recognition of the need for, and an ability to engage in life-long learning.
*	j	A knowledge of contemporary issues.
**	k	An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

BS CSE Program Outcomes

Course Contribution		Program Outcome
***	a	an ability to apply knowledge of computing, mathematics including discrete mathematics as well as probability and statistics, science, and engineering;
**	b	an ability to design and conduct experiments, as well as to analyze and interpret data;
*	c	an ability to design, implement, and evaluate a software or a software/hardware system, component, or process to meet desired needs within realistic constraints such as memory, runtime efficiency, as well as appropriate constraints related to economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability considerations;
	d	an ability to function on multi-disciplinary teams;
**	e	an ability to identify, formulate, and solve engineering problems;
	f	an understanding of professional, ethical, legal, security and social issues and responsibilities;
	g	an ability to communicate effectively with a range of audiences;
	h	an ability to analyze the local and global impact of computing on individuals, organizations, and society;
	i	a recognition of the need for, and an ability to engage in life-long learning and continuing professional development;
*	j	a knowledge of contemporary issues;
**	k	an ability to use the techniques, skills, and modern engineering tools necessary for practice as a CSE professional;
***	l	an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
**	m	an ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;
*	n	an ability to apply design and development principles in the construction of software systems of varying complexity.

Prepared by: Gagan Agrawal