Knight Foundation School of Computing and Information Sciences

Course Title: Operating Systems Principles **Date:** 04/08/2024

Course Number: COP 4610

Number of Credits: 3

Subject Area: Computer Organization	**Subject Area Coordinator:** Dong Chen **Email:** dochen@cs.fiu.edu
Catalog Description: Operating systems design principles and implementation techniques. Address spaces, system call interface, process/threads, inter-process communication, deadlock, scheduling, memory, virtual memory, I/O, file systems.	
Textbooks: Operating System Concepts, 10th Edition Silberschatz, Galvin, and Gagne Wiley (ISBN: 1119800366)	
References:	
Prerequisites Courses: COP 4338 and (CDA 3102 or CDA 4101)	
Corequisite Courses: None	

Type: Required for CS Major

Prerequisites Topics:

- CPU, cache, memory organization
- Instruction set architecture
- Multithreading
- Fundamental data structures

Course Outcomes:

1. Analyze the functions and structures of operating systems.
2. Manage Processes and Threads Effectively.
3. Apply Synchronization and Deadlock Handling Techniques.
4. Evaluate Memory Management Principles.
5. Describe File-system and Storage implementation and I/O subsystem structure.
6. Understand concepts of security and protection in operating systems

Knight Foundation School of Computing and Information Sciences
COP 4610 Operating Systems Principles

Association between Student Outcomes and Course Outcomes

BS in Computing: Student Outcomes	Course Outcomes
1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.	1, 2
2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.	2, 3, 4
3) Communicate effectively in a variety of professional contexts.	
4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.	2, 3

Program Specific Student Outcomes

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. [CS]	2, 3
6) Apply security principles and practices to maintain operations in the presence of risks and threats. [CY]	6
6) Use systemic approaches to select, develop, apply, integrate, and administer secure computing technologies to accomplish user goals. [IT]	6

Assessment Plan for the Course and how Data in the Course are used to assess Student Outcomes

Student and Instructor Course Outcome Surveys are administered at the conclusion of each offering, and are evaluated as described in the School's Assessment Plan: https://abet.cis.fiu.edu/

Knight Foundation School of Computing and Information Sciences
COP 4610 Operating Systems Principles

Outline

Topic	Number of Lecture Hours	Outcomes

• Overview
  ○ Operating system history
  ○ Computer-system organization
  ○ Operating-system structure
	6	1

• Process management
  ○ Processes
  ○ Threads
  ○ CPU scheduling
  ○ Process synchronization.
  ○ Deadlock
	15	2, 3, 6

• Memory management
  ○ Main Memory
  ○ Virtual memory
	9	4, 6

• Storage management
  ○ Mass-storage structure
  ○ I/O sub-systems
	3	5

• File System
  ○ File-system interface
  ○ File-system implementation
	3	5, 6

Knight Foundation School of Computing and Information Sciences
COP 4610 Operating Systems Principles

Performance Measures for Evaluation

Assignment	Percentage of Final Grade
Homework	25%
Exams	50%
Projects	25%

Letter Grade Distribution Table

Letter	Range%		Letter	Range%		Letter	Range%
A	95 or above		B	83 - 86		C	70 - 76
A-	90 - 94		B-	80 - 82		D	60 - 69
B+	87 - 89		C+	77 - 79		F	59 or less

Knight Foundation School of Computing and Information Sciences
COP 4610 Operating Systems Principles

Description of Possible Project

Topic: Operation on processes using system calls in C.

Description: Write a C program that simulates a parallel task execution scenario using process forking. The program should create a specified number of child processes, each performing a unique task. The parent process should wait for all child processes to complete before displaying the final result.

Rubric:

Criteria	Good (100)	Average (60)	Poor (20)
Process Creation	Processes are well-designed and implemented with clear responsibilities and communication channels.	Processes are created, but with some ambiguity in responsibilities or communication.	Processes are poorly defined, leading to confusion and inefficiency.
Task Execution	Tasks are executed efficiently and effectively, meeting or exceeding requirements.	Tasks are completed, but with some delays or minor issues.	Tasks are poorly executed, resulting in significant delays or incomplete work.
Synchronization	Synchronization mechanisms are implemented effectively to manage concurrent operations.	Synchronization mechanisms are implemented, but with occasional issues or inefficiencies.	Synchronization mechanisms are lacking or poorly implemented, leading to race conditions or deadlocks.
Error Handling	Errors are handled gracefully and appropriately, minimizing disruption to the system.	Error handling is present, but some errors may be overlooked or not handled optimally.	Error handling is inadequate, leading to system instability or data corruption.
Code Quality and Readability	Code is well-structured, organized, and follows best practices for readability and maintainability.	Code is readable but may lack consistency or could be better organized.	Code is poorly structured, difficult to read, and may contain numerous flaws.
Correct output	Output meets all specified requirements and expectations.	Output is mostly correct but may contain some minor discrepancies or errors.	Output is incorrect or significantly deviates from the expected results.
Report	A comprehensive report is provided, detailing the process, challenges, solutions, and outcomes.	A report is provided, but may lack depth or thoroughness in addressing all aspects of the project.	The report is incomplete, poorly structured, or missing key information.