CSCI 4015: Design Principles of Operating System

2022 Fall Session	
Total Class Sessions: 30	**Instructor: Staff**
Class Sessions Per Week: 6	**Classroom: TBA**
Total Weeks: 5	**Office Hours: TBA**
Class Session Length (Minutes): 120	**Language: English**
Credit Hours: 4	

Course Description:

This course is an undergraduate level operating system course, designed to provide a comprehensive examination of the design principles of operating systems. Topics include concepts of operating systems, processes and process management, threads and threads programming, CPU scheduling, synchronization and deadlock, memory management and virtual memory, file system, I/O system, and various advanced topics such as networking, distributed systems and OS security. Upon completion of the course, each student will gain a solid understanding of modern operating systems via a series of challenging mini-projects.

Prerequisite: Data Structures and Algorithm Analysis, Computer System Organization

Learning objectives:

Upon completion of this course, students are able to:

- Understand the functionality, design principles and trade-off issues of operating system;
- Define processes and threads, describe different ways to communicate between processes and threads, and apply mutual exclusion-based solutions to synchronize multi-threaded processes without deadlock occurring;
- Identify different scheduling algorithms and their suitability for different types of applications, including compute-bound, I/O-bound and real time;
- Explain the concept of virtual memory, the rationale for on-demand paging, and the role of working sets to avoid thrashing in a caching-based memory hierarchy;
- Demonstrate understanding of fundamental concepts in file system design, including linked and indexed file allocation, mounting, a virtual file system layer, memory mapping, journaling, and performance optimizations for storage media (magnetic and solid state);
- Describe basic concepts to secure and protect operating systems;
- Explain basic concepts in networked operating systems design, including layered network architecture and distributed file systems structure;
- Describe the basic concept of a virtual machine and different types of virtual machines;

- Successfully modify, add functionality to, and re-compile the kernel of an operating system.

Course Materials:

Textbook:

Operating System Concepts, 10th Edition, by Abraham Silberschatz, Peter Baer Galvin, Greg Gagne,Wiley

The 9th Edition is also acceptable

Recommended Text:

Computer Systems: A Programmer's Perspective, 3rd Edition, by Bryant, Randal, and O'Hallaron, David, Pearson

Course Format and Requirements:

The course will take place in a computer lab and the course format including lecture, recorded demos, programming project, and in-class discussion.

The specific topics that will be covered in the classes are listed in the course syllabus. The class period will consist of an active learning environment. During a majority of the class time, students will be actively working on problems under the instructor's guides.

Attendance:

Attendance will not be taken but is strongly recommended. Each student will have three allowed absences and no grade deduction will be made for the first three absences. More than three unexcused absences will result in an automatic reduction in your participation grade, for instance from A- to B+. Your active participation in the class is expected and encouraged.

Course Assignments:

Quizzes:

There will be 5 quizzes. Quizzes will be multiple choice questions. There will be NO make-ups for quizzes for any reason. All of the quizzes will be closed book.

Individual Take-home Programming Exercises

This is a programming intensive course. To succeed in this course, each student are required to take most of their after-class learning time to be engaged in programming exercises and projects.

Individual programming exercises will be assigned at the end of each chapter. It is designed to practice students' programming skills. It is generally in the format of mini-project and can be a crucial component of the group programming project.

Weekly Group Programming Projects

The weekly hands-on programming projects are based on course need. Generally, weekly group programming will be in the format of pair programming. The projects aim to enrich students' knowledge on application of learned concepts, programming languages and programming paradigms. The score will be given based on the correctness of the program. It will count for 40% of your grade for the course.

Midterm Exam:

The in-class, close-book and non-cumulative midterm exam will be given through this course. The midterm exam will be based on the knowledge covered in class. No excuse will be accepted if students do not have legitimate excuses for absence. Physician Statement is required for missing the exam due.

Final Exam:

The final will be in-class, cumulative and close-book. The final exams will be based on concepts covered in class. Note that the final will not be taken during the normal class times. Exact time and location for final will be announced later.

Course Assessment:

Quizzes	5%
Individual Take-home Programming Exercises	20%
Weekly Group Programming Projects	40%
Midterm Exam 1	10%
Midterm Exam 2	10%
Final Exam	15%
Total	**100%**

Grading Scale (percentage):

A+	A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F
98-100	**93-97**	**90-92**	**88-89**	**83-87**	**80-82**	**78-79**	**73-77**	**70-72**	**68-69**	**63-67**	**60-62**	**<60**

Academic Integrity:

Students are encouraged to study together, and to discuss lecture topics with one another, but all other work should be completed independently.

Students are expected to adhere to the standards of academic honesty and integrity that are described in the Chengdu University of Technology's *Academic Conduct Code.* Any work suspected of violating the standards of the *Academic Conduct Code* will be reported to the Dean's Office. Penalties for violating the *Academic Conduct Code* may include dismissal from

the program. All students have an individual responsibility to know and understand the provisions of the *Academic Conduct Code*.

Special Needs or Assistance:

Please contact the Administrative Office immediately if you have a learning disability, a medical issue, or any other type of problem that prevents professors from seeing you have learned the course material. Our goal is to help you learn, not to penalize you for issues which mask your learning.

Tentative Course Schedule:

Week	Class	Topics
Part 1: Overview(Class 1-3)		
1	1	Chapter 1: Introduction
	2	Chapter 2: Operating System Structures
	3	Introduction to Linux Kernel Modules
Part 2: Process Management(Class 4-9)		
1	4	Chapter 3: Processes
	5	Chapter 3: Interprocess Communication
	6	Chapter 4: Threads
2	7	Chapter 4: Concurrency
	8	Chapter 5: CPU Scheduling
	9	Chapter 5: CPU Scheduling
	10	**Midterm 1: Operating System Structures; Process Management**
Part 3: Process Synchronization(Class 11-13)		
2	11	Chapters 6 and 7: Synchronization(Tools and Examples)
	12	Chapters 6 and 7: Synchronization(Tools and Examples)
3	13	Chapters 8: Deadlock
Part 4: Memory Management(Class 14- 16)		
3	14	Chapters 9: Main Memory
	15	Chapters 10: Virtual Memory
	16	Buffer Overflow Lab
Part 5: Storage Management(Class 17-18)		
3	17	Chapters 11: Mass Storage Structure
	18	Chapters 12: I/O Systems
4	**19**	**Midterm 2: Process Synchronization; Memory Management;**

		Storage Management
Part 6: File Systems(Class 20-22)		
4	20	Chapters 13: File-System Interface
	21	Chapters 14: File-System Implementation
	22	Chapters 15: File-System Internals
Part 7: Security and Protection(Class 23-24)		
4	23	Chapters 16/17: Security and Protection
	24	Capabilities Lab
Part 8:Advanced Topics(Class 25-29)		
5	25	Chapters 18: Virtual Machines
5	26	Chapters 19: Networks
5	27	Chapters 19: Distributed Systems
5	28	Parallel Processing Lab
5	29	Introduction to i-cloud computing
5	30	**Warp-up, Q&A, Preparation for Final**
Final Exam Comprehensive, TBA		