Week	Lecture (Zoom)	Tutorial (Zoom or in-person)	Lab (Zoom or in-person)	Deadlines
1	L0: Course Admin L1: Introduction			
2	L2: Process Management (Part I)			
3	L3: Process Management (Part II)	T1: Process Management	B1: Advanced C Mechanisms	
4	L4: Process Scheduling	T2: Process System Calls	B2: A1 due!	Sat, 2pm - Lab Assignment 1 (A1)
5	L5: Process Alternatives and Inter-process Communication	T3: Process Scheduling	B3: A2	
6	L6: Synchronization	T4: Threads and IPC	B4: A2	
Recess				Wed, 2pm - Lab Assignment 2 (A2)
7	Midterm	T5: Synchronization	B5: A2	Wed, 2pm – Midterm
8	L7: Memory Management - Basic	T6: Discussion of Midterm	B5: A3	
9	L8: Memory Management - Disjoint Memory Allocation	T7: Continuous Memory Schemes	B6: A3	
10	L9: Memory Management - Virtual	T8: Disjoint Memory Schemes	B7: A3	Wed, 2pm - Lab Assignment 3 (A3)
11	L10: File System - Introduction	T9: Virtual Memory	B8: A4	
12	L11: File System - Implementation	T10: File System Abstraction	B10: A4	
13	L12: Revision	T11: File System Implementation	B11: A4	Fri, 2pm - Lab Assignment 4 (A4)
Reading				
Exam	Fri, 26 Nov, 2:30pm			Fri, 26 Nov, 2:30pm - Exam

Description

This module introduces the basic concepts in operating systems and links it with contemporary operating systems (eg. Unix/Linux and Windows). It focuses on OS structuring and architecture, processes, memory management, concurrency and file systems. Topics include kernel architecture, system calls, interrupts, models of processes, process abstraction and services, scheduling, review of physical memory and memory management hardware, kernel memory management, virtual memory and paging, caches, working set, deadlock, mutual exclusion, synchronisation mechanisms, data and metadata in file systems, directories and structure, file system abstraction and operations, OS protection mechanisms, and user authentication.

Learning Outcomes

After this course, you should be able to:

- understand how an OS manages computational resources for multiple users and applications, and the impact on application performance
- appreciate the abstractions and interfaces provided by OS
- write multi-process/thread programs and avoid common pitfalls such as deadlocks, starvation and race conditions
- write system programs that utilizes POSIX syscall for process, memory and I/O management
- self-learn and explore advanced OS topics

Grading policy (changes are possible before 9 Aug 2021):

- 10% - Tutorial (quizzes)
- 30% - Lab assignments 1-4
- 20% - Midterm (pen-and-paper with cheat-sheet)
- 40% - Final Exam (pen-and-paper with cheat-sheet)