Dhaka International University

Department of Computer Science and Engineering

Course Code: CSE-310	**Course Title:** Operating System Lab
Course Type (Core Course/Electives/ GED): Core Course	
Year /Semester: 9th Semester	**Academic Session:** 2021-2022
Course Teacher: Md. Shariful Islam	**Credit Value:** 1
Contact Hours: 2 hours	**Total Marks:** 50

Rationale: This lab complements the operating systems course. Students will gain practical experience with designing and implementing concepts of operating systems such as system calls, CPU scheduling, process management, memory management, file systems and deadlock handling using C language in a Linux environment.

Course Objectives: The objectives of this course are-

1. To provide an understanding of the design aspects of operating system concepts through simulation
2. To simulate and implement operating system concepts such as scheduling, deadlock management, file management and memory management
3. To introduce basic Unix commands, system call interface for process management, interprocess communication and I/O in Unix

Course Content:

Overview of Operating Systems (OS): Functionalities and Characteristics of OS, Hardware Concepts Related To OS, CPU States, I/O Channels, Memory Hierarchy, Microprogramming.
The Concept of a Process: Operations on Processes, Process States, Concurrent Processes, Process Control Block, Process Context, UNIX Process Control and Management, PCB, Signals, Forks and Pipes. Interrupt Processing, Operating System Organization, OS Kernel FLIH, Dispatcher.
CPU Scheduling: I/O Burst Cycle, Context Switching, Scheduling, Short Term, Long Term, Scheduling Criteria, Algorithms, First Come First Serve, Shortest Job First, Priority Scheduling, Round Robin.
Process Synchronization: Critical Section Problem, Mutual Exclusion, Races, Two Process Solutions, Bakery Algorithm, Synchronization Hardware, Test and Set, Swap, Semaphores, Deadlocks and Starvation, Classic Synchronization Problems, Readers/Writers, Dining Philosophers.
Deadlocks: System Model, Necessary Conditions for A Deadlock, Mutual Exclusion, Hold and Wait, No Preemption, Circular Wait, Resource Allocation Graphs, Handling Deadlocks, Prevention, Avoidance, Bankers Algorithm.
Memory Management: Address Binding, Compile Time, Load Time, Execution Time, Logical Versus Physical Address Space, Swapping, Contiguous Allocation, Single Partition, Multiple Partition, First Fit, Best Fit, Worst Fit, Internal and External Fragmentation, Paging and Virtual Memory, Demand Paging, Page Replacement, Page Replacement Algorithms, FIFO, Belady's Anomaly, LRU, MFU, Thrashing.
File Organization: Blocking and Buffering, File Descriptor, Directory Structure, File and Directory Structures, Blocks and Fragments, Directory Tree, File Descriptors, UNIX File Structure.

Course Learning Outcomes (CLOs): After learning this course, students will be able to-

1. Design solutions for overcoming the problem of limited resources (software/ hardware) using various algorithms and synchronization libraries.
2. Create shell program for process and file system management
3. Apply synchronized programs using multithreading concepts and simulate CPU Scheduling Algorithms like FCFS, Round Robin, SJF, etc.
4. Implement operating system functionalities using modern tools.

Mapping of Course Learning Outcomes with Program Learning Outcomes:

CLOs	PLOs	Bloom's Taxonomy (Domain/Level)	Teaching-Learning Strategy	Assessment Strategy
CLO1	PLO-C	Cognitive/Apply	Lecture, Demonstration	Q/A, Lab Report
CLO2	PLO-C	Psychomotor/Manipulation	Lecture, Demonstration, Rapport Building, Group Work	Q/A, Lab Report
CLO3	PLO-A	Psychomotor/Precision	Lecture, Demonstration, Group Work	Q/A, Project Work/Report
CLO4	PLO-E	Affective/Responding	Lecture, Demonstration, Group Work	Presentation, Project Work/Report

Mapping of Course – PLO – K – EP/EA

Course	Program Learning Outcomes (PLOs)												Knowledge Profiles								Complex Eng. Problems (EP)							Complex Eng. Activity (EA)				
	Eng. Knowledge	Problem Analysis	Design	Investigation	Modern tool	Engineer & Society	Envir. & Sustainability	Ethics	Teamwork	Communication	Proj. Manage. & Fin.	Life-long learning	Natural Sciences	Mathematics	Eng. Fundamentals	Eng. Specialist	Eng. Design	Eng. Practice	Comprehension	Research Literature	Depth of knowledge (K3-K6, K8)	Range of conflicting requirements	Depth of analysis (no obvious solution)	Familiarity (infrequent issues)	Applicable codes (outside problem)	Stakeholder invol. (Outside group)	Interdependence (many components)	Range of resources	Level of interaction	Innovation	Consequences for society & envir.	Familiarity
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**																				
	C				**CP**	**CA**		**A**	**A**	**PA**	**CA**	**A**																				
	K1 – K4		**K5**	**K8**	**K6**	**K7**							**K1**	**K2**	**K3**	**K4**	**K5**	**K6**	**K7**	**K8**												
	EP (P1 + two or more P2-P7)									EA											**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**A1**	**A2**	**A3**	**A4**	**A5**
CSE-310 (Operating System Lab)	×		×		×										×	×	×	×			×	×	×	×								

Mapping Course Learning Outcomes (CLOs) with the Teaching-Learning and Assessment Strategy:

Week	Topic	Teaching Learning Strategy	Assessment Strategy	Corresponding CLOs
1.	Write a C program to simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time FCFS	● Demonstration ● Discussion	● Demonstration ● Lab reports	CLO3 CLO4
2.	Write a C program to simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time SJF	● Demonstration ● Discussion	● Demonstration ● Lab reports	CLO3 CLO4
3.	Write a C program to simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time RR	● Demonstration ● Discussion	● Demonstration ● Lab reports	CLO3 CLO4
4.	Write a C program to simulate a multi-level queue scheduling algorithm considering the following scenario All the processes in the system are divided into two categories – system processes and user processes System processes are to be given higher priority than user processes	● Case study ● Group work	● Lab reports ● Viva	CLO1 CLO2
5.	Write a C program to simulate the following file allocation strategies Sequential Indexed	● Demonstration ● Case study	● Demonstration ● Quiz tests	CLO1 CLO3

6.	Write a C program to simulate the MVT and MFT memory management techniques	● Rapport building ● MMP (Multimedia Projector) ● Discussion	● Lab reports ● Viva	CLO2 CLO3
7.	Write a C program to simulate the following contiguous memory allocation techniques Worst-fit Best-fit	● Group work ● Case study	● Demonstration ● Quiz tests ● Viva	CLO2 CLO3
8.	Write a C program to simulate the paging technique of memory management.	● MMP (Multimedia Projector) ● Rapport building	● Quiz tests ● Lab reports ●	CLO1 CLO2
9.	Write a C program to simulate the following file organization techniques Single level directory Two-level directories	● Demonstration ● Case study ● Group work	● Demonstration ● Viva	CLO1 CLO2
10.	Write a C program to simulate Banker's algorithm for deadlock avoidance.	● Rapport building ● MMP (Multimedia Projector)	● Lab reports ● Viva ● Quiz tests ●	CLO2 CLO3
11.	Write a C program to simulate disk scheduling algorithms FCFS and SCAN	● Demonstration ● Discussion	● Demonstration ● Lab reports	CLO3
12.	Write a C program to simulate page replacement algorithms FIFO LRU	● Group work ● Rapport building	● Quiz tests ● Viva	CLO3 CLO4

References:

1. Tanenbaum, A., "Operating Systems Design and implementation", Prentice-Hall, 3rd Edition 2006.
2. Crowley, C., "Operating Systems: A Design-Oriented Approach", Irwin, 2002.
3. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, "Operating System Concepts", Addison-Wesley, 6th Edition, 2001.