MODULE DESCRIPTION FORM

نموذج وصف المادة الدراسية

Module Information معلومات المادة الدراسية					
Module Title	**Operating Systems**			**Module Delivery**	
Module Type	**Core**			☒ **Theory** ☒ **Lecture** ☒ **Lab** ☐ **Tutorial** ☐ **Practical** ☐ **Seminar**	
Module Code	**CYBT203**				
ECTS Credits	**4**				
SWL (hr/sem)	**100**				
Module Level			**Semester of Delivery**	3	
Administering Department	Department of Cybersecurity Technical Engineering		**College**	Technical Engineering College of Computer and Artificial Intelligence - Kirkuk	
Module Leader			**e-mail**	leader@ntu.edu.iq	
Module Leader's Acad. Title	Assistant Professor		**Module Leader's Qualification**	Ph.D.	
Module Tutor			**e-mail**		
Peer Reviewer Name	Reviewer Name		**e-mail**		
Scientific Committee Approval Date	2/11/2026		**Version Number**	1.0	

Relation with other Modules العلاقة مع المواد الدراسية الأخرى			
Prerequisite module	Data Structures & Computer Architecture	**Semester**	3 & 4
Co-requisites module	None	**Semester**	-

Module Aims, Learning Outcomes and Indicative Contents أهداف المادة الدراسية ونتائج التعلم والمحتويات الإرشادية	
Module Aims أهداف المادة الدراسية	This module aims to introduce students to the core concepts, structures, and mechanisms of modern operating systems. It focuses on how an operating system manages hardware resources (CPU, memory, storage, and I/O devices) and provides system services to application software. The course aims to provide a deep understanding of process management, CPU scheduling, inter-process communication, concurrency, deadlocks, memory virtualization, and file system implementations.
Module Learning Outcomes مخرجات التعلم للمادة الدراسية	Understand the fundamental structure, components, and functions of modern operating systems. Comprehend process lifecycle, process states, context switching, and multithreading concepts. Analyze and evaluate CPU scheduling algorithms (FCFS, SJF, Priority, Round Robin). Understand synchronization problems, race conditions, semaphores, mutexes, and deadlock handling strategies. Master main memory management, paging, segmentation, and virtual memory page replacement techniques. Understand file system organization, directory structures, disk scheduling, and I/O management. Implement system calls and process control operations using C/C++ in Linux environments.
Indicative Contents المحتويات الإرشادية	The module will cover the following key areas: Overview of Operating Systems Architecture & System Calls Process Concept, Process Control Blocks (PCB), and Process Creation Multithreading Models & POSIX Threads Programming CPU Scheduling Algorithms and Performance Metrics Process Synchronization, Critical Section Problem, Mutex & Semaphores Deadlock Characterization, Prevention, Avoidance (Banker's Algorithm), and Detection Memory Management: Paging, Segmentation, and Translation Lookaside Buffer (TLB) Virtual Memory: Demand Paging, Page Faults, and Page Replacement Algorithms (FIFO, LRU, Optimal) File System Interface, Structure, and Allocation Methods Mass-Storage Structure & Disk Scheduling Algorithms (FCFS, SCAN, C-SCAN)

Learning and Teaching Strategies استراتيجيات التعلم والتعليم	
Strategies	The module will employ a combination of strategies including: Lectures: To provide theoretical foundations on OS concepts, architecture, and algorithms. Group Discussions: To facilitate understanding through the exchange of technical concepts. Problem-Solving Sessions: To apply scheduling, deadlock, and page replacement algorithms practically.

	Practical Lab Sessions: Hands-on experience with Linux CLI, C programming, system calls (fork, exec, wait), and thread synchronization. Tutorials: To delve into complex topics like virtual memory address translation and disk scheduling. Mid-term and final exams: To assess understanding and application of knowledge. Preparatory Week: To give students time to revise and clarify any difficulties before the final exam.

Student Workload (SWL)

الحمل الدراسي للطالب محسوب لـ ١٥ أسبوعا

Structured SWL (h/sem) الحمل الدراسي المنتظم للطالب خلال الفصل	48	**Structured SWL (h/w)** الحمل الدراسي المنتظم للطالب أسبوعيا	8
Unstructured SWL (h/sem) الحمل الدراسي غير المنتظم للطالب خلال الفصل	52	**Unstructured SWL (h/w)** الحمل الدراسي غير المنتظم للطالب أسبوعيا	5
Total SWL (h/sem) الحمل الدراسي الكلي للطالب خلال الفصل	100		

Module Evaluation

تقييم المادة الدراسية

		Time/Number	Weight (Marks)	Week Due	Relevant Learning Outcome
Formative assessment	**Quizzes**	2	5% (5)	5, 10	LO #1, 3 and 5
	Assignments	2	5% (5)	2, 12	LO #2, 4 and 6
	Projects / Lab.	1	10% (10)	Continuous	LO #7
	Report	1	10% (10)	13	LO #3, 5 and 7
Summative assessment	**Midterm Exam**	2 hr	20% (20)	8	LO #1 - 4
	Final Exam	2hr	50% (50)	16	All
Total assessment			100% (100 Marks)		

Delivery Plan (Weekly Syllabus)

المنهاج الاسبوعي النظري

	Material Covered

Week 1	Introduction to Operating Systems: Concept, History, Operating System Structure, and System Calls.
Week 2	Operating System Structures: Monolithic, Layered, Microkernel, and Virtual Machines.
Week 3	Process Concept: Process Control Block (PCB), Process States, Context Switching, and IPC Concept.
Week 4	Threads & Multithreading: Threading models, User vs. Kernel threads, and POSIX Pthreads.
Week 5	CPU Scheduling Concepts: Preemptive vs. Non-preemptive, FCFS, SJF, and SRTF algorithms.
Week 6	Advanced CPU Scheduling: Priority Scheduling, Round Robin (RR), and Multilevel Queue Scheduling.
Week 7	Process Synchronization: Critical Section Problem, Race Conditions, Hardware Support, and Peterson's Solution.
Week 8	Mid-term Exam (الفصل منتصف اختبار) & Synchronization Tools (Mutex Locks & Semaphores).
Week 9	Classic Synchronization Problems: Bounded-Buffer, Readers-Writers, and Dining Philosophers Problems.
Week 10	Deadlocks: Deadlock Characterization, Resource Allocation Graphs, Prevention, and Avoidance (Banker's Algorithm).
Week 11	Main Memory Management: Address Binding, Logical vs Physical Address Space, Swapping, and Contiguous Allocation.
Week 12	Paging & Segmentation: Page tables, Translation Lookaside Buffer (TLB), and Memory Protection.
Week 13	Virtual Memory: Demand Paging, Page Fault Handling, Page Replacement Algorithms (FIFO, Optimal, LRU).
Week 14	File Systems Interface & Implementation: File Concepts, Access Methods, Directory Structure, and Allocation Methods.
Week 15	Mass-Storage Management & Disk Scheduling: FCFS, SSTF, SCAN, C-SCAN, and I/O Hardware Interface.
Week 16	**Preparatory week before the final Exam**

Delivery Plan (Weekly Lab. Syllabus) المنهاج الاسبوعي للمختبر	
	Material Covered
Week 1	Lab 1: Introduction to Linux Environment, Basic Commands (file management, permissions, process monitoring).
Week 2	Lab 2: C Programming in Linux (GCC Compiler) & System Calls (fork(), exec(), wait(), getpid()).

Week 3	Lab 3: Multithreading Programming using POSIX Threads (pthread_create, pthread_join).
Week 4	Lab 4: Simulation and Implementation of CPU Scheduling Algorithms (FCFS & SJF) in C/C++.
Week 5	Lab 5: Simulation of Round Robin and Priority CPU Scheduling Algorithms.
Week 6	Lab 6: Process Synchronization in C using Mutexes and POSIX Semaphores.
Week 7	Lab 7: Revision, System Troubleshooting, and Practical Project Assessment.

Learning and Teaching Resources
مصادر التعلم والتدريس

	Text	Available in the Library?
Required Texts	Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). "Operating System Concepts" (10th ed.). Wiley.	Yes
Recommended Texts	Tanenbaum, A. S., & Bos, H. (2014). "Modern Operating Systems" (4th ed.). Pearson.	Yes
Websites	Yes	

Grading Scheme
مخطط الدرجات

Group	Grade	التقدير	Marks (%)	Definition
Success Group (50 - 100)	**A** - Excellent	امتياز	90 - 100	Outstanding Performance
	B - Very Good	جيد جدا	80 - 89	Above average with some errors
	C - Good	جيد	70 - 79	Sound work with notable errors
	D - Satisfactory	متوسط	60 - 69	Fair but with major shortcomings
	E - Sufficient	مقبول	50 - 59	Work meets minimum criteria
Fail Group (0 – 49)	**FX** – Fail	راسب (قيد المعالجة)	(45-49)	More work required but credit awarded
	F – Fail	راسب	(0-44)	Considerable amount of work required

Note: Marks Decimal places above or below 0.5 will be rounded to the higher or lower full mark (for example a mark of 54.5 will be rounded to 55, whereas a mark of 54.4 will be rounded to 54. The University has a policy NOT to condone "near-pass fails" so the only adjustment to marks awarded by the original marker(s) will be the automatic rounding outlined above.