هيئة تقويم التعليم والتدريب
Education & Training Evaluation Commission

2023
TP-153

Course Specification

(Bachelor)

Course Title: Operating Systems	
Course Code: CSC 1304	
Program: Bachelor in Computer Science	
Department: Computer Science	
College: Faculty of Computers and Information Technology	
Institution: University of Tabuk	
Version: 1.0	
Last Revision Date: 27 July 2022	

Table of Contents

A. General information about the course:

1. Course Identification

1. Credit hours: (3)

2. Course type

A.	☐ University	☒ College	☐ Department	☐ Track	☐ Others
B.	☒ Required			☐ Elective	

3. Level/year at which this course is offered: (Level 6/Year 3)

4. Course general Description:

The course provides a comprehensive and up-to-date coverage of the concepts and principles of Operating Systems (OS) as low-layer software that manages all of the computer's software and hardware communications. The course aims to demystify some concepts, terminology, and algorithms related to Operating Systems design and implementation, including processes, memory management, virtual memory, page replacement algorithms, system calls, interrupts, resource allocation, protection, synchronization, deadlock, scheduling, queuing, and process communication.

5. Pre-requirements for this course (if any):

Computer Organization and Assembly Programming (CSC 1202)

6. Co-requisites for this course (if any):

N/A

7. Course Main Objective(s):

- Identify the main components of the operating systems, including processes, threads, memory management, virtualization, interrupts, system calls, synchronization, deadlock, CPU scheduling, layering, communication, protection, etc., as applied to operating systems design and implementation.

- Understand the software architecture of the operating systems, the interaction between its subsystem components, and services provided by the operating systems as a low-level intermediate layer between the hardware and user applications.
- Recognize the difference between operating systems, distributions, and different supported hardware architectures and time-sharing systems.
- Understand how to perform essential administrative tasks, including configuring, compiling, and installing a new custom kernel along with CentOS distribution.

2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	45	100%
2	E-learning		
3	Hybrid ● Traditional classroom ● E-learning		
4	Distance learning		

3. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	**Lectures**	45
2.	**Laboratory/Studio**	0
3.	**Field**	0
4.	**Tutorial**	0
5.	**Others (specify)**	0
Total		45

B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Code of PLOs aligned with program	Teaching Strategies	Assessment Methods
1.0	Knowledge and understanding			
1.1	Understand the importance of the operating systems and identify the main components of the operating systems, such as processes, threads, scheduling, files, memory, virtual memory, sharing mechanisms, IPC, deadlock, layering, system calls, interrupts, protection in OS, etc.	**K1**	Lectures Class Discussions Case Study	Exams Assignments
1.2	Understanding the software architecture, complexity, modularity, and principal roles of the Operating Systems focusing on Linux Systems and x86 architecture. Also, the ability to compare them to different embedded or general-purpose OS on other architectures like ARM.	**K2**	Lectures Class Discussions	Exams Assignments
1.3	Recognize essential algorithms, e.g., process scheduling and memory management algorithms.	**K3**	Lectures Class Discussions	Exams Assignments

Code	Course Learning Outcomes	Code of PLOs aligned with program	Teaching Strategies	Assessment Methods
1.4	Understand the development process of Linux Kernel and install the latest mainline/stable kernel. Getting familiar with Linux kernel environment as a free, open-source, monolithic, and multitasking Operating System (CentOS).	**K3**	Lectures Class Discussions	Exams Assignments
2.0	**Skills**			
2.1	Ability to perform administrative tasks on Linux Systems "CentOS or any."	**S1,S2**	Lectures Case Study	Exams Assignments Projects
2.2	Capability to perform manual configurations, compile and install for Linux kernel on x86 architecture, system debugging, "distribution can vary," and trace Linux kernel code, specifically x86 architecture.	**S3**	Lectures Class Discussions Case Study	Exams Assignments Projects
...				
3.0	**Values, autonomy, and responsibility**			
...				

C. Course Content

No	List of Topics	Contact Hours
1.	Introduction to Operating Systems: Organization, interrupts, modern components, user mode to kernel mode, computing environments, and open-source OS	3
2.	Operating-System Structures, I/O Systems, Virtual Machines (part 1): OS services, system calls, monolithic kernel, layers, microkernel, modular, and hybrid design.	3
3.	Operating-System Structures, I/O Systems, Virtual Machines (part 2): Boot process, performance monitoring, kernel modules.	3

4.	Introduction to Linux Kernel Development: configure, compile, install, tracing and code reading.	3
5.	Processes and Threads: concept, creation, structure, communication, termination	3
6.	CPU Scheduling Algorithms (part 1): Concepts, cycle, illusion, burst-time, scheduling decisions, preemptive and non-preemptive.	3
7.	CPU Scheduling Algorithms (part 2): Race conditions, switching context, ready queue, waiting queue, jobs queue, dispatcher, Arrival Time, Completion time, burst time, turnaround time, waiting time, response time, Scheduling Criteria and Goals.	3
8.	CPU Scheduling Algorithms (part 3): Long-term scheduler, Medium-term scheduler, Short-term scheduler, Scheduling Policies (FCFS, SJF, RR, Priority, Multiple Queues, CFS, Real-Time policies), Convoy Effect, Starvation, Aging.	3
9.	Process Synchronization: Critical-Section Problem, Synchronization hardware support, Mutex Locks, Semaphores, Monitors, Liveness.	3
10.	Deadlocks: System Model, Deadlock Characterization, Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.	3
11.	Memory Management (part 1): Measurement Units, Storage Hierarchy, data migration, Base and Limit Registers, Hardware Address Protection, Address Binding (Compile time, load time, execute time), logical and physical address space, and memory Hierarchy Management.	3
12.	Memory Management (part 2): Memory allocation contiguous and non-contiguous techniques, single contiguous memory, multiple portioning, paging, and segmentation.	3
13.	Memory Management (part 3): Paging, Page fault, Cache Line, TLB, virtual memory, demand paging, and page replacement algorithms (FIFO, LRU, Optimal).	3
14.	Mass Storage: Disk Structure, Disk Scheduling Algorithms.	3
15.	File Systems: File Concept, Access Methods, Directory Structure, Allocation Methods, Free Space Management.	3
Total		**45**

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Class Works (Two Homework (5% + 5%) and Project 20%) *(Individual Assignments)*	1 ~ 13	30%
2.	Two Mid-Term Exams (15% +15%)	7 and 12	30%
3.	Final Exam	16	40%
…			

*Assessment Activities (i.e., Written test, oral test, oral presentation, group project, essay, etc.).

E. Learning Resources and Facilities

1. References and Learning Resources

Essential References	Operating System Concepts Tenth Edition. by Avi Silberschatz, Peter Baer Galvin, Greg Gagne. John Wiley & Sons, Inc. ISBN 978-1-118-06333-0
Supportive References	“Mastering Linux Kernel Development,” A kernel developer's reference manual by Raghu Bharadwaj
Electronic Materials	Saudi Digital Library (SDL) (www.sdl.edu.sa)
Other Learning Materials	Blackboard Platform

2. Required Facilities and equipment

Items	Resources
facilities (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	Classroom or lab - 32 seats.
Technology equipment (projector, smart board, software)	Data show, internet, smart board, VM, etc.
Other equipment (depending on the nature of the specialty)	N/A

F. Assessment of Course Quality

Assessment Areas/Issues	Assessor	Assessment Methods
Effectiveness of Teaching	Faculty, Program Leaders, and Advisory Board	Both Direct and Indirect
	Students	Indirect
Effectiveness of Students Assessment	Faculty, Program Leaders, Advisory Board, and Independent Opinion	Both Direct and Indirect
Quality of Learning Resources	Faculty, Students, and Advisory Board	Indirect
The Extent to which CLOs	Faculty, Program Leaders, Advisory	Direct (as in section B)

Assessment Areas/Issues	Assessor	Assessment Methods
have been Achieved	Board, and Independent Opinion	and Indirect/Surveys
	Students	Indirect
Other	-	-

Assessors (Students, Faculty, Program Leaders, Peer Reviewer, Others (specify)

Assessment Methods (Direct, Indirect)

G. Specification Approval

COUNCIL /COMMITTEE	
REFERENCE NO.	
DATE	