MULTIMEDIA UNIVERSITY OF KENYA
Riding on Technology, Inspiring Innovation

Faculty of Computing and Information Technology

2025/2026 First Semester [Sept. 1 to Dec. 19, 2025]

Operating Systems – II

Contacts

Lecturer: Patrick Mokodir

Tel. Ext. 2078 (by appointment)

Email: pmokodir@mmu.ac.ke

Consultation hours:

Mondays: 12:30 – 1:30pm and Wednesdays: 12:30 – 1:30pm

Purpose
To enable the student understand the principles and applications of operating systems.

Learning outcomes

By the end of this course unit the student should be able to:

1. Explain the concept of interprocess communication and coordination
2. Describe message passing algorithms
3. Describe distributed file systems, security and process management

Rationale
An Operating System is to manage a Computer Hardware and software resources efficiently and provide user-friendly environment. An Operating System is a System Program that controls the execution of application program and acts as an interface between applications and the computer hardware. It also place a curtail role in maintaining system security, protecting data and ensuring that processes do not interfere with one another.
This course enables to learn internal functioning of Operating System and will help in identifying appropriate Operating System for given Application/Task.

Tentative Weekly Learning Schedule

Week	Topic	Content	Remarks
1	Introduction	Basic Terminologies, Distributed OS, Networked OS, Applications.	Priming codes
2	Design and Implementation of operating systems	Objectives, Principles, Strategies, Implementation strategies, platforms	
3	Distributed Processing	Principles of distributed operating systems, Rationale for distributed systems	Task 1 due
4	Distributed Processing	Algorithms for Distributed Processing	
5	Transactions Processing	Transactions properties, architecture, Serializability, Transactions Management	
6	Distributed Concurrency and Deadlocks. Failures	Descriptions, implementation. Deadlock types, Deadlock avoidance Failure and Robustness	Task 2 due
7	Models of Distributed Systems	Host-based, Processor Pool, Workstation, Server and Integrated Models	
8	Transparency	Importance, types, implementation	
9	Naming Service	Implementing a naming service, Static Maps, Broadcasting, Name Servers, Prefix Tables	Task 3 due
10	Distributed Process Management		
11	Remote Procedure Calls	How RPC Works? Remote Procedure Calling Mechanism. Implementation of RPC. Considerations for Usage. Limitations	
12	Distributed Shared Memory	Types, architectures and examples	
13	Distributed File Systems	Types, architectures and examples	Task 4 due
14	Case studies: A comparative study of the operating systems.	Modern OS	
15	Course Review and Evaluation	UNIX, LINUX, UBUNTU Competence	

Reference Books

1. Modern Operating Systems

Grading:

1. Assignments - 20%
2. Mid-term - 10%
3. Final Exam - 70%

NB: Final exam will have both theoretical grounding questions (40%) and actual implementation questions (60%). Final pass grades will be A, B, C or D. Any score below 40% will be graded as failure.

Attendance Policy

a) No partial or technical appearance for lectures.
b) All lectures will begin at the stipulated time unless communicated earlier.
c) Lectures that are to be conducted on holidays October 10 and October 20th will be rescheduled in advance and communicated.
d) Any student who misses two or more lectures will automatically not be graded and no sitting for the final exam.
e) Prior to sitting for exam, your attendance will be tabulated, evaluated and posted to the class group and copied to the CoD for concurrency.

Mode of Lectures

- We will have both physical and virtual classes.

Due to the massive nature of the course, all notes will be sent to students 48 hours before a lecture.

Programming tasks covered in OS-1 [To Be completed before lecture 2 begins]

a) Write a C/Python program to calculate average waiting time and Turnaround Time of n processes with First Come First Serve (FCFS) CPU scheduling algorithm.
b) Write a C/Python program to calculate average waiting time and Turnaround Time of n processes with Shortest Job First (SJF) CPU scheduling algorithm.
c) Write a C/Python program to calculate average waiting time and Turnaround Time of n processes with Priority CPU scheduling algorithm.
d) Write a C/Python program to calculate average waiting time and Turnaround Time of n processes with Round Robin (RR) CPU scheduling algorithm.
e) Write a C/Python program to implement Banker's Algorithm.
f) Write a C/Python program on Least Recently Used (LRU) Page Replacement algorithm.
g) Write a C/Python program on sequential file allocation method.