Pokhara University
Faculty of Science and Technology

Course Code: CMP 232	Full Marks: 100
Course title: Operating Systems (3-1-2)	Pass Marks: 45
Nature of the Course: Theory and Practical	Total Lectures: 45 hours
Level: Bachelor	Program: BE

1. Course Description

This course is designed to encompass the fundamental concepts of operating system. These concepts include the system software, internal construct of operating system, run application software and perform various tasks. This course also introduces the emerging new trended operating system for real-time, distributed, cloud, mobile systems. After completion of this course, students can design and develop an operating system for any devices.

2. General Objective

- To acquaint the students with basic concepts of resource allocation and management.
- To acquaint the students with structure of operating systems and their functionality.
- To develop the skills in students to select and design optimal resource allocation schedules.
- To acquaint the students with the knowledge of process and thread, I/Os, Memory, CPU, disk management.
- To acquaint the students with basic concepts of Operating systems new trends such as real-time, distributed, cloud, and mobile systems.

3. Methods of Instruction

Lecture, Discussion, Readings, Practical works and Project works.

4. Contents in Detail

Specific Objectives	Contents
• Familiarize with basic concepts of Operating systems, and its architecture. • Understanding the successive evolution of operating system	**Unit 1: Introduction** [4 hrs.] 1.1 Concept and function of operating systems 1.2 Operating systems concept and functionality - Processes, Files, System Calls, The Shell 1.3 Operating System Structure - Monolithic System, Layered, Virtual Machines, Client-server. Microkernel and exokernel 1.4 Types and Evolution of Operating Systems
• Familiarize with Task, Process and threads • Implement of resource allocation techniques • Understanding the mutual exclusion for resource utilization	**Unit 2: Processes and Threads** [10 hours] **2.1 Process** - Definition, states and transition diagram - Process control block (PCB) - Concurrent and Parallel processes **2.2 Interprocess Communication and Synchronization** - Introduction, Race Condition, Critical Regions and condition, Avoiding critical region: Mutual Exclusion and Serializability, - Mutual exclusion conditions. mutual exclusion:

	disabling interrupts, lock variable, strict alteration (Dekker's algorithms, Peterson's algorithms), The TSL instruction, sleep and wakeup, producer and consumer problem - Types of mutual exclusion (Semaphore, Monitors, Bounded buffer, Message passing), - Classical IPC Problems (The Dining Philosophers problem, The Readers and Writers problem, The Sleeping Barber Problem) - Serializability: Locking Protocols and Time Stamp Protocols **2.3 Process Scheduling** - Basic Concept, Type of scheduling (Preemptive scheduling, Non-preemptive scheduling, Batch, Interactive, Real time scheduling), - Scheduling criteria and performance analysis, scheduling algorithm with examples (First come first served, Shortest-job-first, Round-robin, Shortest process next, Shortest remaining time next, Real time, Priority fair share, guaranteed, Lottery scheduling) **2.3 Threads** - Definitions of Threads, Types of thread process (Single and multithreaded process), Benefits of Multithreading - Threads Models (Many-to-one model, One-to-one Model, Many-to many model) - User Space and Kernel Space Threads, - Difference between Processes and Threads
• Conceptualize the role and working procedure of memory • Familiarizing with virtual memory management • Understanding the page replacement algorithms	**Unit 3: Memory Management** [10 hours] - Introduction: Storage organization, Memory hierarchy - Storage allocation, Contiguous verses non-contiguous store allocation, logical and physical memory - Fragmentation, fixed partition and variable partition for multiprogramming, Logical versus physical address space - Relocation and Protection - Memory management with swapping: Memory management with bitmaps and linked list, Memory management without swapping, - Contiguous-memory allocation: memory protection, memory allocation, Fragmentation (Inter fragmentation and external fragmentation) - Paging, Structure of page table: Hierarchical page table, Hashed page table, Inverted page table, Shared page table, - Virtual memory- Introduction, Paging, Page Table, Block mapping, Direct mapping, Translation Look Aside Buffers) Demand

	paging, Thrashing - Page replacement, Page replacement algorithms: First-in-first-out, Not recently used, Optimal page replacement, Second chance page replacement, Least Recently used, Clock page replacement, Working set page replacement, WS clock page replacement - Segmentation, Segmentation with paging - Coalescing and Compaction
• Conceptualization of Kernel and its role in system software	**Unit 4: Kernel** [2 hrs] 1. Introduction, Architecture of the Kernel, Types of Kernels 2. Context Switching (Kernel and User mode), Kernel implementation processes
• Understand the role of input/output devices • Understand the different approaches for optimal output	**Unit 5: Input/Output Management** [5 hrs] - Introduction, Interrupts Handlers - Principles of I/O Hardware (I/O Device, Device Controller, Memory Mapped I/O, Direct Memory Access) - Principles of I/O Software (Goals of I/O Software, Polled I/O verses Interrupt Driven I/O, Character User Interface and Graphical User Interface, Device Driver, Device Independent I/O Software, User -space I/O Software, - System Resources: Preemptable and Non-preempable, Method of handling Deadlocks, Deadlock prevention, - Deadlock avoidance: Banker's Algorithm, Deadlock detection: Resource allocation graph, Recovery from Deadlock - Redundant Array of Inexpensive Disks - RAM Disks
• Understand the file and filing mechanism • Familiarize with directory and its management techniques	**Unit 6: File Systems** [3 hrs] - File and File Organization (Blocking and Buffering, File Descriptor, file Naming, File Structure, File Types, File Attributes, File Operations, File Access Methods) - Directories Management (Single-level directory systems, Hierarchical Directory systems, Path names, Directory operation) - Access Methods: Sequential, Direct, other access methods, Protection: Types of access, Access Control List, - File System Implementation: Contiguous allocation, Linked list allocation, linked list allocation using an Index nodes - Security and Multi-media files
• Familiarize with current trends of operating systems	**Unit 7: New Trends in Operating System** - Concept, character and role of - Real-time Operating System, - Distributed Operating System

	- Cloud Operating System - Mobile Operating System - Security issues and method of deployment - Memory wall and bottleneck for Operating system

5. Practical Works

Laboratory work of 12 hours should cover the operating system such as MS Windows, Linux, etc. Students should complete the following tasks in laboratory:

1. Installation and user, application management in Windows (current version)
2 Simulation of Process Scheduling Algorithms using C/C++
3. Simulation of Disk Arm Scheduling Algorithms
4. System Administration (user, disk, role, etc.) in any open-source operating system.

6. List of Tutorials

The various tutorial activities that suit your course should cover all the content of the course to give students a space to engage more actively with the course content in the presence of the instructor. Students should submit tutorials as assignments or class works to the instructor for evaluation. The following tutorial activities of 10 hours as per the convenient of student and faculty.

A. Review and Question/Answer-based Tutorials: (10 hrs)
   a. Case study on any Open-source operating system.
   b. Students ask questions within the course content, assignments and review key course content in preparation for tests or exams.

7. Evaluation System and Students'
Responsibilities Evaluation System

The internal evaluation of a student may consist of assignments, attendance, internal assessment, lab reports, project works etc. The internal evaluation scheme for this course is as follows:

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		30	Semester-End examination	50
Attendance & Class Participation	10%			
Assignments	20%			
Presentations/Quizzes	10%			
Internal Assessment	60%			
Practical		20		
Attendance & Class Participation	10%			
Lab Report/Project Report	20%			
Practical Exam/Project Work	40%			
Viva	30%			
Total Internal		50		
Full Marks: 50 + 50 = 100				

Student Responsibilities

Each student must secure at least 45% marks separately in internal assessment and practical evaluation with 80% attendance in the class in order to appear in the Semester End Examination. Failing to get such a score will be given NOT QUALIFIED (NQ) to appear for the Semester-End Examinations. Students are advised to attend all the classes, formal exam, test, etc. and complete all the assignments within the specified time period. Students are required to complete all the requirements defined for the completion of the course.

8. Prescribed Books and References

Text Books

1) 1. Andrew s. Tanenbaum, *"Distributed Operating System*", Pearson

References

1) Andrew s. Tanenbaum, *"Modern Operating System",* PHI, 3rd Ed. 2011
2) A. Silberschatz, P.B. Galvin, G. Gagne *"Operating System Concepts*", Wiley, 8th Ed.
3) D M Dhamdhere, "*System Programming and Operating System*"- Tata McGraw-Hill, 20