

OPERATING SYSTEMS[BCS303]

Contents

- Course Information
- Introduction to Operating System
- Course Objectives
- Course Outcome
- Overview of syllabus

Course Information

- Course Type: IPCC [Integrated Professional Core Course]
- No of credits:04
- Teaching Hours
Theory: 03
Lab:02
- CIE Marks:50
- SEE:Marks:50
- Total Marks :100



Course Information contd..

- CIE for the theory component of the IPCC (maximum marks 50)
- IPCC means practical portion integrated with the theory of the course.
- CIE marks for the theory component are 25 marks and that for the practical component is 25 marks.
- 25 marks for the theory component are split into 15 marks for two Internal Assessment Tests (Two Tests each of 15 Marks with 01-hour duration, are to be conducted) and 10 marks for other assessment methods

Course Information contd..

■ CIE (Theory)

Components	Marks	
Test-1	40	Scaled down to 15 marks
Test-2	40	
Test-3	40	
Assignments	10	Scaled down to 10 marks
Activities	10	
MOOC course	10	
Total	25	

- The student has to secure 40% of 25 marks to qualify in the CIE of the theory component of IPCC.

Course Information contd..

■ CIE (Practical)

Components	Marks	
Conduction of Experiments - Observation book - Record - Execution of experiment - Viva	15	Average marks of all experiments
Test	50	Scaled down to 10 marks
Total	25	



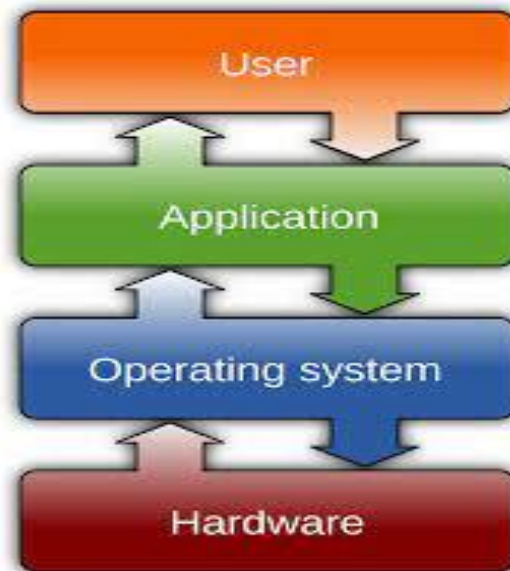
Course Information contd..

SEE

- The question paper will have ten questions. Each question is set for 20 marks.
- There will be 2 questions from each module. Each of the two questions under a module (with a maximum of 3 sub-questions), **should have a mix of topics** under that module.
- The students have to answer 5 full questions, selecting one full question from each module.
- Marks scored by the student shall be proportionally scaled down to 50 Marks
- **The theory portion of the IPCC shall be for both CIE and SEE, whereas the practical portion will have a CIE component only. Questions mentioned in the SEE paper may include questions from the practical component.**

Introduction to Operating System

- The operating system (OS) is a fundamental software component that manages and controls the basic operations of a computer system.
- Its role is to act as an intermediary between the hardware and software





Examples of Operating systems



chromeOS



ubuntu



iOS

symbian
OS



UNIX

www.knowcomputing.com



Key roles of Operating System

1. Process Management.
2. Memory Management.
3. File Systems Management.
4. Device Management.
5. User Interface
6. Security and Protection
7. Networking
8. Error Handling and Recovery



Why study Operating System?

- Computer science is a broad field that encompasses everything from computer systems and networks architecture to cyber security, theory and algorithms, hardware and software.
- Computer scientists may work as software developers, computer systems analysts, database administrators, or other careers related to the way software systems operate.
- Any computing profession will have to use operating systems, so understanding how they work and how they vary is useful to be able to accomplish much of anything with computing.

Course objectives

- To Demonstrate the need for OS and different types of OS
- To discuss suitable techniques for management of different resources
- To demonstrate different APIs/Commands related to processor, memory, storage and file system management.



Overview of syllabus

Modules

Module-1

Introduction to operating systems, System structures: What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Special-purpose systems; Computing environments.

Operating System Services: User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure; Virtual machines; Operating System debugging, Operating System generation; System boot.

Module-2

Process Management: Process concept; Process scheduling; Operations on processes; Inter process communication

Multi-threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues.

Process Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Thread scheduling: Multiple-processor scheduling



Overview of syllabus contd..

Modules

Module-3

Process Synchronization: Synchronization: The critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization;

Deadlocks: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.

Module-4

Memory Management: Memory management strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation.

Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing.



Overview of syllabus contd..

Modules

Module-5

File System, Implementation of File System: File system: File concept; Access methods; Directory and Disk structure; File system mounting; File sharing; Implementing File system: File system structure; File **system implementation**; Directory implementation; Allocation methods; Free space management. Secondary Storage Structure, **Protection:** Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Protection: Goals of protection, Principles of protection, Domain of protection, Access matrix.

Course outcomes

At the end of the course, the student will be able to:

- CO 1. Explain the structure and functionality of operating system
- CO 2. Apply appropriate CPU scheduling algorithms for the given problem.
- CO 3. Analyze the various techniques for process synchronization and deadlock handling.
- CO 4. Apply the various techniques for memory management
- CO 5. Explain file and secondary storage management strategies.
- CO 6. Describe the need for information protection mechanisms

Textbooks

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 8th edition, Wiley-India, 2015

Reference Books

1. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition
2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw- Hill, 2013.
3. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE), 2014.
4. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson.

ANY
QUESTIONS



Thank You!



PRACTICAL COMPONENT OF IPCC*(May cover all / major modules)*

1. Develop a c program to implement the Process system calls (fork (), exec(), wait(), create process, terminate process)
2. Simulate the following CPU scheduling algorithms to find turnaround time and waiting time
 - a) FCFS
 - b) SJF
 - c) Round Robin
 - d) Priority.
3. Develop a C program to simulate producer-consumer problem using semaphores.
4. Develop a C program which demonstrates interprocess communication between a reader process and a writer process. Use mkfifo, open, read, write and close APIs in your program.

PRACTICAL COMPONENT OF IPCC *contd...*

5. Develop a C program to simulate Bankers Algorithm for DeadLock Avoidance.
6. Develop a C program to simulate the following contiguous memory allocation Techniques:
 - a) Worst fit b) Best fit c) First fit.
7. Develop a C program to simulate page replacement algorithms:
 - a) FIFO b) LRU
8. Simulate following File Organization Techniques
 - a) Single level directory b) Two level directory



PRACTICAL COMPONENT OF IPCC *contd...*

9. Develop a C program to simulate the Linked file allocation strategies.
10. Develop a C program to simulate SCAN disk scheduling algorithm.