# PAPER VII: OPERATING SYSTEMS

**COMPREHENSIVE QUESTION BANK**

Strictly formatted into 1, 2, 5, and 8 Mark distribution per unit.

## UNIT I: Introduction, Architecture, Structures, and Systems

### *1 Mark Questions (Short Answer / Objective)*

**Q1.** What is an Operating System?

**Q2.** Define the term 'kernel' in an operating system context.

**Q3.** What is the primary objective of a batch processing system?

**Q4.** Define multi-programming.

**Q5.** What is a time-sharing operating system?

**Q6.** State the main difference between a hard real-time system and a soft real-time system.

**Q7.** What is a system call?

**Q8.** Name the mechanism used to pass parameters to the operating system during a system call via registers.

**Q9.** What are system programs?

**Q10.** Define dual-mode operation in a computer system architecture.

**Q11.** What is the purpose of the mode bit?

**Q12.** What is a bootstrap program and where is it stored?

**Q13.** Define a monolithic operating system structure.

**Q14.** What is the core principle of a microkernel architecture?

**Q15.** State one advantage of a layered approach to operating system design.

**Q16.** What is an interrupt?

**Q17.** Define the term 'system architecture'.

**Q18.** What is the role of an interrupt vector?

**Q19.** Give an example of a user-operating system interface.

**Q20.** What is a distributed operating system?

### *2 Mark Questions (Brief Explanations)*

**Q1.** Differentiate between multi-programming and time-sharing systems.

**Q2.** Explain the concepts of symmetric and asymmetric multiprocessing.

**Q3.** What is the role of system calls? Give two examples of standard system calls.

**Q4.** Explain the concept of dual-mode operation (User Mode and Kernel Mode).

**Q5.** Why is a modular or microkernel architecture preferred over a monolithic structure?

**Q6.** Describe the function and purpose of a command interpreter or shell.

**Q7.** What are the major services provided by an operating system to users?

**Q8.** Explain how the hardware protects memory using base and limit registers.

**Q9.** What is a real-time operating system (RTOS)? List two common applications.

**Q10.** Differentiate between tightly-coupled and loosely-coupled computer systems.

**Q11.** Explain the separation of mechanism and policy in operating system design.

**Q12.** What are system programs? How do they differ from system calls?

**Q13.** Describe the direct memory access (DMA) architecture and its purpose.

**Q14.** What is an interrupt service routine (ISR) and how is it invoked?

**Q15.** Explain the concept of a virtual machine at a high level.

### *5 Mark Questions (Analytical / Descriptive)*

**Q1.** Explain the detailed architecture of a computer system highlighting how CPUs and I/O devices interact.

**Q2.** Discuss the layered approach to operating system structure with its structural merits and demerits.

**Q3.** Describe the microkernel architecture of an operating system. How does it improve reliability and extensibility?

**Q4.** Explain the step-by-step process of how an interrupt is handled by the hardware and software.

**Q5.** Discuss the different categories of system calls provided by an operating system with examples for each.

**Q6.** Elaborate on batch processing, multi-programming, and time-sharing operating systems.

**Q7.** Explain the relationship between system calls, system programs, and user applications with a structural diagram description.

**Q8.** Discuss the design and implementation challenges and goals when designing a modern operating system.

**Q9.** Explain dual-mode operation in detail and explain why it is essential for system protection and stability.

**Q10.** Describe real-time systems, comparing their processing constraints, design, and practical use cases.

### *8 Mark Questions (Comprehensive Essay / Numerical Cases)*

**Q1.** Provide an exhaustive analysis of operating system structures, comparing Monolithic, Layered, Microkernel, and Modules architectures. Discuss their individual trade-offs in terms of performance and maintenance.

**Q2.** Explain computer system architecture and execution. Detail the boot process from power-on (bootstrap code execution) to loading the kernel, initializing the system, and moving to user space.

**Q3.** Write a detailed essay on system calls, illustrating how a user program invokes a system call (e.g., reading data from a file). Detail how the control passes through the system call interface into the kernel and returns.

**Q4.** Compare and contrast batch operating systems, multi-programmed systems, time-sharing networks, and real-time platforms. Detail how resource management and CPU utilization strategies differ in each architecture.

**Q5.** Discuss the design and implementation of operating systems. Explain how separating policies from mechanisms helps in configuring, modifying, and scaling modern kernels across diverse computing environments.

## UNIT II: Process Management, Threads, CPU Scheduling, and Deadlocks

## ***1 Mark Questions (Short Answer / Objective)***

**Q1.** What is a process?

**Q2.** List the primary states of a process in its lifecycle.

**Q3.** What is a Process Control Block (PCB)?

**Q4.** Define context switching.

**Q5.** What is a thread?

**Q6.** Name the three multithreading models commonly used.

**Q7.** What is the primary goal of CPU scheduling?

**Q8.** Define CPU utilization as a scheduling criterion.

**Q9.** What scheduling algorithm is based on assigning a fixed time slice to each process?

**Q10.** What is turnaround time?

**Q11.** Define starvation in the context of CPU priority scheduling.

**Q12.** What is a deadlock?

**Q13.** State the four necessary conditions for a deadlock to occur.

**Q14.** What does a resource-allocation graph with a cycle indicate in a single-instance system?

**Q15.** Name one algorithm used for deadlock avoidance.

**Q16.** What is Inter-Process Communication (IPC)?

**Q17.** Define a zombie process.

**Q18.** What is the function of the short-term scheduler?

**Q19.** Name the two models of IPC.

**Q20.** What is a safe state in deadlock avoidance?

## ***2 Mark Questions (Brief Explanations)***

**Q1.** Differentiate between a process and a program.

**Q2.** Explain the components or fields stored inside a Process Control Block (PCB).

**Q3.** What is context switching? State its primary performance overhead.

**Q4.** Compare user-level threads and kernel-level threads.

**Q5.** Explain the Many-to-One and One-to-One multithreading models.

**Q6.** Differentiate between short-term, medium-term, and long-term schedulers.

**Q7.** Define the following scheduling criteria: Throughput and Response Time.

**Q8.** Explain the Convoy Effect observed in First-Come, First-Served (FCFS) scheduling.

**Q9.** What is aging, and how does it solve the problem of starvation?

**Q10.** Explain the concept of cooperative vs. preemptive CPU scheduling.

**Q11.** Describe the four necessary conditions that must hold simultaneously for a deadlock.

**Q12.** What is the difference between deadlock prevention and deadlock avoidance?

**Q13.** Briefly explain the role of a safe state in the Banker's Algorithm.

**Q14.** Explain message passing and shared memory models of Inter-Process Communication.

**Q15.** What is a critical section, and what are the requirements for its solution?

### *5 Mark Questions (Analytical / Descriptive)*

**Q1.** Describe the lifecycle states of a process with a complete state transition diagram.

**Q2.** Explain the concept of Inter-Process Communication (IPC) and discuss the shared memory and message-passing mechanisms.

**Q3.** Discuss the benefits of multithreaded programming and explain the structural differences in multithreading models.

**Q4.** Explain the Round Robin (RR) CPU scheduling algorithm with its benefits, criteria, and the impact of time quantum choices.

**Q5.** Describe Multilevel Queue and Multilevel Feedback Queue scheduling mechanisms with appropriate application scenarios.

**Q6.** Discuss the four necessary conditions for deadlock and describe strategies to systematically eliminate or prevent each condition.

**Q7.** Explain the Banker's Algorithm for deadlock avoidance, defining safety criteria, allocation states, and request criteria vectors.

**Q8.** Given processes with burst times, calculate average waiting and turnaround times under Shortest Job First (SJF) scheduling.

**Q9.** Compare and contrast First-Come, First-Served (FCFS) and Preemptive Priority CPU scheduling algorithms.

**Q10.** Discuss deadlock detection strategies for systems with multiple instances of resource types and how recovery is managed.

### *8 Mark Questions (Comprehensive Essay / Numerical Cases)*

**Q1.** Provide a rigorous analysis of CPU scheduling algorithms. Evaluate FCFS, SJF (Preemptive and Non-Preemptive), Priority, and Round Robin scheduling based on throughput, response time, and waiting criteria with dry-run case equations.

**Q2.** Examine deadlocks in detail. Explain the four necessary conditions and write a thorough guide on the implementation of the Banker's Algorithm for deadlock avoidance, detailing safety checks and resource request sequences.

**Q3.** Discuss the complete mechanics of processes and threads. Detail process lifecycle transitions, state contents inside a PCB, context-switching overheads, thread structure, and the performance differences between user and kernel threads.

**Q4.** Analyze Inter-Process Communication and synchronization. Critically compare shared-memory systems against message-passing frameworks, and evaluate the critical section problem along with the conditions required for a valid solution.

**Q5.** Detail the structure and execution of Multilevel Feedback Queue (MLFQ) scheduling. Explain how processes move between queues based on execution history, and discuss how deadlocks can be recovered via process termination or resource preemption.

## UNIT III: Memory Management Strategies and Virtual Memory

## *1 Mark Questions (Short Answer / Objective)*

**Q1.** What is logical address space?

**Q2.** Define physical address space.

**Q3.** What is the primary function of a Memory Management Unit (MMU)?

**Q4.** What is swapping in memory management?

**Q5.** Define internal fragmentation.

**Q6.** Define external fragmentation.

**Q7.** What is paging?

**Q8.** What does a page table map?

**Q9.** Define a page frame.

**Q10.** What is segmentation?

**Q11.** Define virtual memory.

**Q12.** What is demand paging?

**Q13.** What is a page fault?

**Q14.** State the goal of an optimal page replacement algorithm.

**Q15.** What is thrashing?

**Q16.** What causes Belady’s Anomaly?

**Q17.** Name the page replacement technique that uses the history of the recent past.

**Q18.** What is a translation lookaside buffer (TLB)?

**Q19.** What is a dirty bit or modify bit in page tables?

**Q20.** Define the working-set model.

## *2 Mark Questions (Brief Explanations)*

**Q1.** Differentiate between logical and physical addresses.

**Q2.** Explain internal fragmentation and external fragmentation with a brief example.

**Q3.** Describe the function of the Translation Lookaside Buffer (TLB) in a paging system.

**Q4.** What is the difference between paging and segmentation?

**Q5.** Explain dynamic loading and dynamic linking.

**Q6.** What is a page fault? Outline the immediate hardware response.

**Q7.** Explain Belady's Anomaly and name an algorithm that suffers from it.

**Q8.** Describe the concept of demand paging.

**Q9.** What is thrashing? State its primary symptom inside the CPU.

**Q10.** Explain the difference between contiguous and non-contiguous memory allocation.

**Q11.** How does compaction eliminate external fragmentation?

**Q12.** What is the purpose of the valid/invalid bit in a page table entry?

**Q13.** Briefly explain the Least Recently Used (LRU) page replacement principle.

**Q14.** What is the purpose of swapping in a time-shared operating system?

**Q15.** Define the concept of pure demand paging.

### *5 Mark Questions (Analytical / Descriptive)*

**Q1.** Explain the core concept of paging along with the hardware mechanisms needed to translate a logical address to a physical address.

**Q2.** Discuss the segmentation memory management scheme, describing its logical structure, segment tables, and protection strategies.

**Q3.** Compare contiguous memory allocation schemes, specifically detailing First-Fit, Best-Fit, and Worst-Fit allocation strategies.

**Q4.** Explain the concept of virtual memory management and discuss its benefits regarding system utilization and multi-programming.

**Q5.** Describe the complete sequence of events that occurs when a page fault happens in a demand-paged system.

**Q6.** Given a reference string, compute page faults using FIFO and LRU page replacement algorithms with a 3-frame layout trace.

**Q7.** Explain the Optimal Page Replacement algorithm and discuss why it is difficult to implement in real operating systems.

**Q8.** Discuss thrashing in detail, describing its causes, visual performance drops, and strategies used to prevent or bound it.

**Q9.** Explain the working-set model and how it utilizes locality of reference to prevent system thrashing.

**Q10.** Describe structural enhancements to page tables, including hierarchical paging, inverted page tables, and hashed page tables.

### *8 Mark Questions (Comprehensive Essay / Numerical Cases)*

**Q1.** Provide a comprehensive analysis of paging. Detail address translation using page tables, TLB hardware implementation, hit ratios, and calculate effective memory access times under multi-level paging schemes.

**Q2.** Examine virtual memory and demand paging. Trace every step of a page fault lifecycle, from the hardware trap and disk I/O operation to page table updates and instruction restart sequences.

**Q3.** Evaluate page replacement algorithms. Trace the execution of a long reference string under FIFO, LRU, and Optimal policies. Demonstrate Belady's Anomaly using FIFO and prove why stack-based algorithms avoid it.

**Q4.** Compare and contrast contiguous memory allocation against non-contiguous options (Paging and Segmentation). Detail how each strategy handles internal and external fragmentation, protection bits, and logical sharing.

**Q5.** Write a detailed technical paper on thrashing. Explain the relationships between multi-programming levels, CPU utilization, and thrashing. Elaborate on the Working-Set model and Page-Fault Frequency strategies to counter it.

## UNIT IV: Storage Management, Disk Scheduling, and File Systems

### *1 Mark Questions (Short Answer / Objective)*

**Q1.** What is a file in an operating system?

**Q2.** Name the attribute that identifies a file uniquely within a directory.

**Q3.** What is sequential file access?

**Q4.** Define direct or relative file access.

**Q5.** What is a directory structure?

**Q6.** Name the most flexible directory structure used in modern OS design.

**Q7.** What is the purpose of file sharing?

**Q8.** Define access control lists (ACLs) in file protection.

**Q9.** What is disk scheduling?

**Q10.** What does seek time mean in disk storage?

**Q11.** Define rotational latency.

**Q12.** Which disk scheduling algorithm behaves like an elevator?

**Q13.** What is the unique characteristic of the C-SCAN scheduling algorithm?

**Q14.** What does RAID stand for?

**Q15.** Which RAID level provides pure striping without redundancy?

**Q16.** What is mirroring in RAID technology?

**Q17.** Define file protection.

**Q18.** What is a mounting point in a file system?

**Q19.** Name the data structure that represents a file inside UNIX systems.

**Q20.** What is disk formatting?

## *2 Mark Questions (Brief Explanations)*

**Q1.** Explain sequential access and direct access methods for files.

**Q2.** Differentiate between single-level and two-level directory structures.

**Q3.** What is an acyclic-graph directory? State its primary advantage.

**Q4.** Explain the difference between disk seek time and rotational latency.

**Q5.** Describe the fundamental goals of file protection mechanisms.

**Q6.** Explain the concept of disk striping in RAID structures.

**Q7.** Compare the FCFS and SSTF disk scheduling objectives.

**Q8.** What is the 'elevator algorithm' in disk storage management?

**Q9.** How does C-SCAN improve upon standard SCAN disk scheduling?

**Q10.** Explain the difference between a physical disk structure and a logical file layout.

**Q11.** What information is typically stored within a file control block (FCB)?

**Q12.** What is parity information in RAID systems?

**Q13.** Describe access rights managed via file protection bits in UNIX.

**Q14.** What are the problems associated with sharing files across networked environments?

**Q15.** Briefly define the LOOK and C-LOOK disk scheduling variants.

## *5 Mark Questions (Analytical / Descriptive)*

**Q1.** Explain file concepts, detailing common file attributes, types, operations, and file internal structures.

**Q2.** Discuss different directory systems, focusing on Tree-Structured and Acyclic-Graph directory layouts with clear diagrams.

**Q3.** Describe the access verification methods used for file protection, comparing Access Control Lists (ACLs) with capability lists.

**Q4.** Explain the FCFS and SSTF disk scheduling algorithms with their operational steps, benefits, and common drawbacks.

**Q5.** Describe the SCAN and C-SCAN disk scheduling algorithms, detailing how head movement changes performance and wait uniformity.

**Q6.** Discuss RAID technology, outlining its core purposes and detailing RAID levels 0, 1, and 5 with architectural layouts.

**Q7.** Explain file system mounting and file sharing models, highlighting user authentication and permissions in distributed frameworks.

**Q8.** Describe mass-storage structure, explaining physical disks, blocks, tracks, sectors, and logical block mappings.

**Q9.** Compare LOOK and C-LOOK disk scheduling behaviors against SCAN variants, demonstrating efficiency optimizations.

**Q10.** Explain the layer models of a standard file system, tracing operations from logical application inputs down to physical storage blocks.

## *8 Mark Questions (Comprehensive Essay / Numerical Cases)*

**Q1.** Provide a rigorous analysis of disk scheduling algorithms. Given a disk queue request array, compute total head movements using FCFS, SSTF, SCAN, C-SCAN, LOOK, and C-LOOK. Evaluate their individual starvation liabilities.

**Q2.** Write a detailed essay on RAID technology. Discuss performance, reliability, and data reconstruction strategies across RAID levels 0 through 6, and analyze the trade-offs between mirroring, striping, and parity calculations.

**Q3.** Examine the architectural design of a file system. Detail the layers of a file system, directory implementation methods (linear list vs. hash table), and allocation policies (contiguous, linked, and indexed allocations) with structural trade-offs.

**Q4.** Discuss directory structures, file sharing, and security. Critically analyze Single-level, Two-level, Tree-structured, and Acyclic-graph directories. Detail how modern platforms handle concurrent sharing and multi-user access permissions safely.

**Q5.** Detail mass-storage configurations and file access protection systems. Explain disk architecture mapping, bad-block management, the role of swap space, and security matrices such as ACLs and domain-based access controls.