Define an operating system.

What is a system call?

List five services provided by an operating system.

What is system boot?

Mention different operating system structures.

Explain the structure of a computer system.

Distinguish between user mode and kernel mode.

Describe the operations performed by an OS.

What is the role of the shell in an operating system?

Explain types of system calls with examples.

How does system booting work?

What is the difference between a monolithic and microkernel OS?

List the types of system calls available in Unix/Linux.

What is the purpose of abstraction in OS?

Differentiate between layered OS and modular OS.

Illustrate the OS structure with a diagram.

Write a pseudo-code for a system call implementation.

What happens during context switching?

How do user programs interact with the OS?

Explain the purpose of interrupts in OS design.

What is a process?

Define process control block (PCB).

List different types of schedulers.

What is context switching?

Explain process creation steps.

Describe process states using a diagram.

How does the scheduler work in an OS?

What is the difference between processes and threads?

Explain the benefits of multithreading.

Describe the Round Robin scheduling algorithm.

Apply FCFS to a set of processes and compute average waiting time.

Simulate SJF algorithm with an example.

Illustrate how context switching works using a diagram.

Simulate priority scheduling with appropriate data.

Write a code snippet to implement Round Robin.

Draw a Gantt chart for a given process list.

Write pseudo-code to demonstrate multithreading.

Model a process queue using data structures.

Compare preemptive and non-preemptive scheduling.

What is starvation and how is it avoided?

What is a critical section?

Define semaphore.

What are the necessary conditions for a deadlock?

List classical problems of synchronization.

What is Peterson’s algorithm?

Explain race condition with an example.

Describe the producer-consumer problem.

How are semaphores used to solve synchronization issues?

Differentiate between deadlock prevention and avoidance.

Explain monitors with respect to synchronization.

Apply Peterson’s algorithm to two-process coordination.

Implement semaphores in code for mutual exclusion.

Solve the readers-writers problem using semaphores.

Demonstrate deadlock with a resource allocation graph.

Describe the dining philosophers problem.

Implement critical section access using synchronization tools.

Construct a resource allocation graph with potential deadlock.

Use Banker’s algorithm to determine system safety.

Identify the problem of starvation in synchronization.

Analyze different synchronization techniques and their use.

What is paging in memory management?

Define thrashing.

List types of memory allocation.

What is a page fault?

Mention the functions of the Memory Management Unit (MMU).

Describe contiguous memory allocation.

Explain the structure of a page table.

Differentiate paging from segmentation.

What is demand paging?

Explain the concept of copy-on-write.

Calculate page faults using FIFO.

Apply LRU to a reference string and calculate faults.

Solve paging with 3 memory frames and 6 page references.

Simulate first-fit and best-fit algorithms.

Translate logical to physical address in paging.

Replace pages using optimal replacement algorithm.

Implement a simple memory manager in code.

Design a page table for a given memory layout.

Draw the segmentation memory layout.

Write code for frame allocation policy simulation.

Compare paging and segmentation with pros and cons.

What causes internal fragmentation?

What causes external fragmentation?

How does the OS handle thrashing?

Differentiate between global and local page replacement.

Analyze effectiveness of working set model.

Compare memory management strategies in Linux and Windows.

What is a file system?

Define access matrix.

List types of directory structures.

What is disk scheduling?

What are the goals of protection in OS?

Explain file access methods.

Describe the file mounting process.

Differentiate between file protection and system protection.

Explain SCAN and LOOK disk scheduling algorithms.

What is an access control list?

Simulate FCFS disk scheduling with 5 requests.

Use SSTF for scheduling disk requests.

Apply LOOK algorithm to given disk request sequence.

Design an access matrix for a given system.

Implement file permissions in a Linux system.

Create a directory structure diagram.

Construct a scheduling chart using C-SCAN.

Apply protection domains in a multi-user system.

Demonstrate how to set file access permissions.

Compare various disk scheduling techniques.

Rank scheduling algorithms by performance.

Evaluate effectiveness of different protection models.

Assess the performance impact of directory structures.

Review limitations of SCAN disk scheduling.

Compare logical and physical file structures.

Explain process synchronization in real-time systems.

Design a basic OS layout for an embedded system.

Simulate an OS boot sequence diagrammatically.

Write a shell program to demonstrate system call behavior.

Create a basic process scheduler simulator.

Build a memory management simulator in Python.

Design a file access simulator for education use.

Model deadlock prevention strategy in code.

Create a memory allocation comparison tool.

Build a prototype scheduler that compares RR and SJF.

Develop a visualization tool for segmentation.

Design a disk scheduler UI simulation.

Create a mini OS prototype supporting command parsing.

What is the role of the dispatcher in process scheduling?

How do user-level and kernel-level threads differ?

Describe the impact of thread switching on performance.

What is the function of a context in process control?

Define long-term scheduler and short-term scheduler.

Explain the purpose of fork() system call.

Describe exec() system call with example.

What is the wait() system call used for?

How does IPC (Interprocess Communication) work?

Differentiate between pipes and sockets.

How are shared memory and message passing different?

What is a signal in Unix-based OS?

Explain asynchronous vs synchronous communication.

Write a program demonstrating fork-exec-wait system calls.

Create an example for message queue IPC.

Simulate producer-consumer using shared memory.

Illustrate deadlock with circular wait and hold.

How do access rights affect system protection?

What are the principles of protection?

Explain domain-based access control.

What is capability-based security?

Compare discretionary and mandatory access control.

Describe Linux file system security mechanisms.

Explain ACL and its implementation.

How does the OS handle file sharing between users?

Demonstrate domain switching with examples.

Explain how the OS provides isolation.

What is memory-mapped file access?

Describe demand segmentation.

Illustrate segmentation with paging.

How does copy-on-write reduce memory usage?

What are major causes of memory fragmentation?

Write a program to simulate page faults.

Compare FIFO and Optimal page replacement.

Analyze performance of file access methods.

List factors influencing choice of scheduling algorithm.

Define starvation with example.

What is the difference between job and process?

Explain different thread libraries in Unix.

What is a lightweight process?

How do threads share resources?

Discuss multiple processor scheduling.

What is CPU burst and I/O burst?

Define time quantum in Round Robin.

What is turnaround time and waiting time?

Explain preemptive vs non-preemptive in terms of interrupt.

How does system protection contribute to security?

What is a security policy in OS?

How does the OS ensure confidentiality and integrity?

Define trusted computing base (TCB).

Describe access control matrix with example.

How are security breaches detected in OS?

What is a trap and how is it different from an interrupt?

Define soft and hard real-time systems.

What is a virtual machine?

How does a hypervisor work?

What is the role of virtualization in modern OS?

Describe OS support for mobile systems.

Compare different types of real-time OS.

How does OS manage background and foreground processes?

Explain system logs and their role in debugging.

How does journaling work in file systems?

What is inode in Unix file systems?

Describe the bootloader’s function.

What is the difference between BIOS and UEFI?

Explain kernel panic and how OS handles it.

Describe how OS supports system calls in hardware.

What are the benefits of open-source OS?

Compare Linux, Windows, and Mac OS.

What are embedded operating systems?

Describe the structure of Android OS.

How is memory managed in cloud OS?

What is containerization and how is it supported by OS?

What are microservices and how does OS scheduling support them?

Summarize the evolution of operating systems over time.