# OPERATING SYSTEM PLACEMENT PREPARATION NOTES

*Comprehensive Interview & Revision Guide*

## 1. Introduction to Operating Systems

An Operating System (OS) is system software that acts as an intermediary between the user and computer hardware. It manages hardware resources and provides services for computer programs.

### *Core Functions of OS:*

• Process Management • Memory Management • File System Management • Device Management • Security & Protection • I/O System Management

## 2. Process Management

A process is a program in execution. It includes program code, data, and process control information. The OS manages process creation, scheduling, and termination.

### *Process States:*

1. New 2. Ready 3. Running 4. Waiting (Blocked) 5. Terminated

### *Scheduling Algorithms:*

• First Come First Serve (FCFS) • Shortest Job First (SJF) • Round Robin (RR) • Priority Scheduling • Multilevel Queue Scheduling

### *Important Terms:*

• Context Switching: Saving and restoring process state when switching CPU between processes. • Throughput, Turnaround Time, Waiting Time, Response Time.

## 3. Process Synchronization

Synchronization ensures orderly process execution to avoid race conditions when multiple processes access shared data.

### *Key Mechanisms:*

• Semaphores – integer variables used for signaling (wait() and signal()) • Mutex – mutual exclusion lock • Monitors – high-level synchronization constructs

### *Classic Problems:*

• Producer-Consumer Problem • Readers-Writers Problem • Dining Philosophers Problem

## 4. Deadlocks

A deadlock occurs when a set of processes are blocked because each holds a resource and waits for another.

### *Conditions for Deadlock:*

1. Mutual Exclusion 2. Hold and Wait 3. No Preemption 4. Circular Wait

### *Handling Methods:*

• Prevention – avoid at least one of the four conditions • Avoidance – e.g., Banker's Algorithm • Detection & Recovery – detect cycles and terminate or preempt

## 5. Memory Management

Memory management tracks each byte in a computer’s memory and allocates space as needed by processes.

### *Allocation Techniques:*

• Contiguous Allocation – Fixed & Variable partitions (leads to fragmentation) • Paging – divides memory into fixed-size blocks • Segmentation – divides into logical segments (code, data, stack) • Virtual Memory – enables execution of processes larger than physical memory

### *Page Replacement Algorithms:*

• FIFO • LRU (Least Recently Used) • Optimal • Second Chance

## 6. File Systems

The file system organizes data and manages how data is stored and retrieved.

### *File Access Methods:*

• Sequential Access • Direct (Random) Access • Indexed Access

### *File Allocation Methods:*

• Contiguous • Linked • Indexed

### *Disk Scheduling Algorithms:*

• FCFS • SSTF • SCAN / C-SCAN • LOOK / C-LOOK

## 7. I/O Systems

The I/O subsystem manages device communication and data buffering between memory and peripherals.

• I/O buffering, caching, spooling • Device drivers and interrupt handling

## 8. Security & Protection

Security ensures data integrity, authentication, and controlled access.

• Access Matrix, ACL (Access Control List), Capability Lists • Authentication: Passwords, Biometrics • Encryption: Symmetric and Asymmetric

## 9. Modern OS & Virtualization

Modern systems support virtualization, multi-core processing, and cloud-based OS services.

• Virtual Machines & Hypervisors • Containers (Docker Basics) • System Calls (fork, exec, wait) • Multithreading & Concurrency

## 10. Frequently Asked OS Interview Questions

1. Difference between process and thread. 2. What is a race condition? How is it avoided? 3. Explain context switching. 4. What is a deadlock? How can it be prevented? 5. What is the difference between paging and segmentation? 6. Explain working of LRU page replacement. 7. What is a critical section? 8. Explain Banker's Algorithm. 9. What is the difference between preemptive and non-preemptive scheduling? 10. Define starvation and aging. 11. Explain internal and external fragmentation. 12. What are system calls? Give examples. 13. Difference between user-level and kernel-level threads. 14. Explain the purpose of the dispatcher in CPU scheduling. 15. What is virtual memory? 16. Define semaphore and mutex. 17. What are file access methods? 18. Difference between cache and buffer. 19. Explain the concept of thrashing. 20. What is a safe state in deadlock avoidance?

## 11. OS Quick Revision Cheat Sheet

• Process = Program in execution • PCB = Process Control Block • Context Switch = Save/Restore CPU state • Deadlock Conditions = Mutual Exclusion, Hold & Wait, No Preemption, Circular Wait • Virtual Memory = Logical > Physical • Page Replacement = FIFO, LRU, Optimal • Scheduling = FCFS, SJF, RR, Priority • Critical Section = Shared resource code • Semaphore Ops = wait(), signal() • Thrashing = Excessive paging • Fragmentation = Unused memory gaps