# ✅ Complete Operating Systems Interview Topic List

---

## 🧠 1. \*\*Basic OS Concepts\*\*

- What is an Operating System?

- OS Roles and Responsibilities

- Types of OS:

- Batch

- Time-Sharing

- Distributed

- Real-Time (RTOS: Hard vs Soft)

- Networked OS

- Multiprogramming

- Multiprocessing

- Multithreading

- Mobile OS

- Kernel Types:

- Monolithic Kernel

- Microkernel

- Hybrid Kernel

- Exokernel

- System Calls and APIs

- User mode vs Kernel mode

- Trap instructions

---

## 🧠 2. \*\*Processes and Threads\*\*

- Process Concepts:

- Process vs Program

- Process Lifecycle (new → terminated)

- Process Control Block (PCB)

- Thread Concepts:

- Thread vs Process

- Types of Threads:

- User-Level Threads (ULT)

- Kernel-Level Threads (KLT)

- Multithreading Models:

- Many-to-One

- One-to-One

- Many-to-Many

- Context Switching (CPU Save/Restore)

- Fork, exec, wait, exit (Linux)

- Process Hierarchy: Parent, Child, Orphan, Zombie

- Daemon Processes

- Thread Pools

---

## 🧠 3. \*\*CPU Scheduling\*\*

- Preemptive vs Non-Preemptive

- Scheduling Criteria:

- CPU Utilization, Throughput, Turnaround Time, Waiting Time, Response Time

- Scheduling Algorithms:

- FCFS

- SJF (Preemptive & Non-Preemptive)

- Round Robin

- Priority Scheduling

- Multilevel Queue

- Multilevel Feedback Queue

- EDF (Earliest Deadline First)

- Rate Monotonic Scheduling (RTOS)

- Gantt Charts and Scheduling Metrics Calculation

- Starvation and Aging

---

## 🧠 4. \*\*Synchronization & Concurrency\*\*

- Critical Section Problem

- Solutions:

- Peterson’s Algorithm

- Bakery Algorithm

- TestAndSet(), Swap()

- Hardware Support for Synchronization

- Mutex vs Semaphore vs Spinlock

- Counting vs Binary Semaphores

- Monitor

- Busy Waiting vs Blocking

- Thread Safety & Atomicity

- Reentrant Functions

- Race Conditions

- Deadlock:

- Necessary Conditions

- Resource Allocation Graph

- Prevention, Avoidance (Banker’s Algorithm), Detection, Recovery

- Livelock

- Starvation

- Bounded Buffer (Producer-Consumer)

- Reader-Writer Problem

- Dining Philosophers Problem

- Barrier Synchronization

---

## 🧠 5. \*\*Inter-Process Communication (IPC)\*\*

- Shared Memory vs Message Passing

- Pipes (Anonymous, Named/FIFO)

- Signals and Signal Handling

- Message Queues

- Semaphores (System V vs POSIX)

- Sockets (UNIX and Network)

- RPC (Remote Procedure Call)

- Memory-Mapped Files

- Pipes vs FIFO vs Shared Memory

---

## 🧠 6. \*\*Memory Management\*\*

- Logical vs Physical Address

- MMU (Memory Management Unit)

- Swapping

- Contiguous Memory Allocation:

- Fixed & Variable Partitions

- First Fit, Best Fit, Worst Fit

- Fragmentation:

- Internal vs External

- Compaction

- Paging:

- Page Table, Frame, Offset

- TLB (Translation Lookaside Buffer)

- Multi-level Page Tables

- Inverted Page Table

- Hashed Page Table

- Segmentation:

- Segment Table, Segment Fault

- Segmentation with Paging

- Memory Protection and Access Control

- Demand Paging

---

## 🧠 7. \*\*Virtual Memory\*\*

- Virtual Address Space

- Demand Paging and Page Fault

- Copy-on-Write (COW)

- Page Replacement Algorithms:

- FIFO

- LRU

- Optimal

- LFU

- Second-Chance

- Clock

- Working Set Model

- Thrashing

- Belady’s Anomaly

- Prepaging

- TLB Miss Handling

---

## 🧠 8. \*\*File Systems\*\*

- File Concepts:

- Metadata, File Attributes

- File Types and Extensions

- Access Modes: Sequential, Direct, Indexed

- File Operations (Create, Open, Read, Write, Seek, Close)

- File Descriptor Table

- Directory Structure:

- Single-Level, Two-Level

- Acyclic Graph, General Graph

- Mounting and Unmounting

- File Allocation:

- Contiguous

- Linked

- Indexed

- Free Space Management:

- Bitmap

- Linked List

- Grouping

- Inode Structure (UNIX)

- Journaling and Crash Recovery

- Symbolic Links vs Hard Links

- File System Implementation Layers:

- Application Layer

- Logical File System

- File-Organization Module

- Basic File System

- I/O Control

---

## 🧠 9. \*\*I/O Systems\*\*

- I/O Hardware

- Polling vs Interrupt Driven I/O

- Interrupt Handling

- DMA (Direct Memory Access)

- I/O Scheduling:

- FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK

- Disk Structure:

- Cylinders, Platters, Tracks, Sectors

- RAID Levels (0–6)

- Buffering and Caching

- Spooling

---

## 🧠 10. \*\*Security & Protection\*\*

- Goals of Security: Confidentiality, Integrity, Availability

- Access Control:

- ACLs, Capability Lists

- Authentication vs Authorization

- Protection Rings

- Domain of Protection

- Unix Permissions (chmod, umask, suid)

- Trojan Horses, Viruses, Worms, Rootkits

- Encryption (Symmetric, Asymmetric)

- Secure OS Design (e.g., SELinux)

---

## 🧠 11. \*\*Deadlocks (In-Depth)\*\*

- Four Coffman Conditions

- Detection via Wait-For Graph

- Resource Allocation Graphs (Cycles)

- Prevention Techniques:

- Hold and Wait elimination

- Preemption

- Circular Wait ordering

- Banker’s Algorithm (Safety Sequence Check)

- Deadlock Recovery (Kill, Rollback)

---

## 🧠 12. \*\*Linux Internals / System Programming\*\*

- `fork()`, `exec()`, `wait()`, `exit()`

- `getpid()`, `getppid()`

- `kill()`, `signal()`, `sigaction()`

- `nice()` and `renice()`

- `/proc` and `/sys` file systems

- Zombie vs Orphan Processes

- `strace`, `lsof`, `top`, `vmstat`, `free`, `iostat`

- Shell Implementation (bonus)

- Bash Scripting Basics

---

## 🧠 13. \*\*Advanced Topics\*\*

- Memory-Mapped I/O vs Port-Mapped I/O

- NUMA (Non-Uniform Memory Access)

- Kernel Preemption and Latency

- Multi-Core CPU Scheduling

- Cache Coherency and False Sharing

- Lock-Free / Wait-Free Data Structures

- Page Coloring

- IOMMU

- CPU Affinity

- Real-Time Scheduling

- OS Boot Process (BIOS → Bootloader → Kernel)

---

## 🧠 14. \*\*Virtualization and Containers\*\*

- Hypervisor Types:

- Type 1 (Bare Metal) vs Type 2 (Hosted)

- Virtual Machines vs Containers

- Docker Internals

- Namespaces

- Cgroups

- UnionFS (AUFS, OverlayFS)

- KVM, QEMU, Xen

- Container Scheduling & Orchestration (Kubernetes Overview)

- Cloud-native OS behavior

---

## 🧠 15. \*\*Distributed Operating Systems\*\*

- Characteristics and Goals

- Transparency Types (Access, Location, Replication)

- Clock Synchronization (NTP, Berkeley)

- Election Algorithms (Bully, Ring)

- Mutual Exclusion in Distributed Systems

- Google’s Borg, Omega, and Kubernetes Architecture

---

## 🧠 16. \*\*Miscellaneous / System Design-Adjacent\*\*

- Page Cache vs Buffer Cache

- Mmap() vs Read()

- Thread vs Coroutine (Userland schedulers)

- Epoll vs Select vs Poll

- File Descriptor Table Limits

- Load Average (1m, 5m, 15m meaning)

- Swappiness

- Out-of-Memory Killer (OOM)

- Memory Pressure Detection

- Reclaiming Memory (kswapd)

- HugePages, Transparent HugePages

- Memory Leaks and Debugging with Valgrind

- Scheduling Tuning (`/etc/sched\_debug`)

 Diagrams (e.g., Process State Transition, Page Table)

 Code (e.g., fork/exec, semaphore implementations)

 Output samples

 Gantt chart problems

 Real-world scenario Q&A