

Operating Systems - Comprehensive Notes

1. Introduction to Operating Systems

An Operating System (OS) is system software that acts as an intermediary between users and the computer hardware. It provides a stable, consistent environment for applications to execute. The OS handles hardware resources and ensures efficient resource utilization. It simplifies programming by providing high-level interfaces for complex operations.

2. Evolution of Operating Systems

The development of OSs evolved over multiple generations:

- First generation: No OS, manual programming.
- Second generation: Batch processing systems.
- Third generation: Multiprogramming and time-sharing systems.
- Fourth generation: Personal computers with GUI-based OSs.
- Current trends: Distributed, mobile, and cloud-based OSs.

3. Process Management

A process is a program in execution. The OS manages processes using a Process Control Block (PCB). Key functions include:

- Creation and deletion of user and system processes.
- Process scheduling using algorithms like FCFS, SJF, Round Robin, and Priority Scheduling.
- Context switching between processes.
- Interprocess Communication (IPC) via shared memory or message passing.

4. Threads and Multithreading

Threads are lightweight processes. Benefits of multithreading include efficient CPU usage and faster execution. Threads can be:

- User-level or kernel-level.
- Managed using libraries (e.g., POSIX threads) or native OS support.

5. CPU Scheduling

CPU scheduling maximizes CPU utilization and throughput. Algorithms include:

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

Each has its use cases depending on the desired response time and fairness.

6. Memory Management

Memory management involves:

- Tracking memory usage.
- Allocating and deallocating memory space as needed.
- Techniques include paging (dividing memory into fixed-size pages), segmentation (logical division), and virtual memory (logical memory larger than physical memory).

Page replacement algorithms: FIFO, LRU, Optimal Replacement.

7. Deadlocks

A deadlock is a situation where processes wait indefinitely for resources held by each other.

Deadlock handling involves:

- Prevention (avoid at least one of the Coffman conditions)
- Avoidance (e.g., Banker's Algorithm)
- Detection and recovery strategies

8. Storage Management

File systems manage data storage and retrieval. Functions include:

- File creation, deletion, read/write operations
- Directory management
- File allocation methods: contiguous, linked, and indexed allocation

Popular file systems: FAT, NTFS, ext3, ext4

9. I/O Systems

OS handles I/O through device drivers and interrupt handling. Types of I/O:

- Programmed I/O
- Interrupt-driven I/O
- Direct Memory Access (DMA)

The I/O subsystem includes buffering, caching, spooling, and device queuing.

10. Security and Protection

Security protects data from unauthorized access. Mechanisms include:

- Authentication and access control (passwords, biometrics)
- Encryption for confidentiality
- Firewalls and antivirus software
- User roles and permissions

11. Distributed and Cloud Operating Systems

Distributed OSs manage a group of distinct computers and make them appear to users as a single system. Features include resource sharing, concurrency, and fault tolerance. Cloud OSs operate across data centers, managing virtual machines and services.

12. Mobile Operating Systems

Mobile OSs (e.g., Android, iOS) are optimized for touchscreen interfaces, battery efficiency, and real-time responsiveness. They have a different architecture focused on lightweight design and app sandboxing for security.

13. Virtualization

Virtualization allows multiple OSs to run on a single physical machine. A hypervisor (Type 1 or Type 2) creates and manages virtual machines. Benefits include isolation, scalability, and better resource utilization.

14. Case Studies

- Windows: GUI-based, multitasking, uses NTFS.
- Linux: Open-source, command-line driven, supports ext3/ext4.
- macOS: Unix-based, GUI with high focus on design and security.

15. Summary

Operating Systems are crucial for system functionality and user interaction. From managing memory and processes to handling security and file systems, they provide the backbone for modern computing environments.