



Study Guide: Operating Systems

1. What Is an Operating System?

Software that manages hardware, provides abstractions, and enables programs to run safely and efficiently.

Core roles:

- Process management
 - Memory management
 - I/O and device control
 - File systems
 - Security and protection
-

2. Processes & Threads

Process

- A running program with its own memory space.
- States: new → ready → running → waiting → terminated.

Thread

- A lightweight process sharing memory with others.
- Used to run tasks in parallel within one program.

Context Switching

- Saving/restoring registers, program counter.
- Expensive; affects performance.

Scheduling Algorithms

- **FCFS** – first come first served
 - **SJF** – optimal for average wait time
 - **Round Robin** – good for time-sharing
 - **Priority scheduling** – risk of starvation
 - **Multilevel queues** – separate queues by type
-

3. Synchronization & Concurrency

Critical Section Problem

Need to prevent races when threads share data.

Tools

- **Mutexes / Locks**: exclusive access
- **Semaphores**: counters for resource availability
- **Monitors**: combine lock + condition variables

Common Problems

- **Deadlock**: circular wait (4 Coffman conditions)
- **Starvation**: process never scheduled
- **Race conditions**: order of operations affects result

4. Memory Management

Address Spaces

- Logical vs. physical addresses
- Translation done by MMU

Paging

- Divides memory into fixed-size pages + frames.
- Eliminates external fragmentation.
- Page table maps virtual → physical pages.
- **TLB** used for caching page translations.

Segmentation

- Logical division of memory (stack, heap, text).

Virtual Memory

- Uses secondary storage (swap) to extend memory.
- Page replacement algorithms:
 - FIFO
 - LRU
 - Optimal (theoretical baseline)

5. File Systems

Responsibilities:

- Organize files and directories
- Provide permissions and metadata
- Ensure consistency (journaling == crash recovery)

Allocation strategies:

- Contiguous
 - Linked
 - Indexed (used by UNIX)
-

6. I/O Systems

- Device drivers manage hardware.
- Interrupts tell CPU when work is done.
- DMA offloads large data transfers from CPU.