Operating Systems Cheat Sheet

1. Processes vs Threads

- Process: Independent execution unit with separate memory. Higher overhead.
- Thread: Lightweight unit within a process, shares memory. Lower overhead.
- Use threads for parallelism within a program; processes for isolation.

2. Multithreading & Concurrency

- Multithreading: Multiple threads run concurrently in a single process.
- Parallelism: Threads run on separate cores simultaneously.
- Concurrency: Tasks progress at the same time (even on a single core).
- Used in servers, schedulers, parallel processing.

3. Synchronization

- Prevents race conditions in multithreaded environments.
- Mechanisms: Mutex (mutual exclusion), Semaphore (counting), Locks, Atomic operations.
- Race Condition: Two threads updating the same resource without coordination.

4. Memory Management

- Stack: Fast memory for function calls (LIFO).
- Heap: Slower, dynamic memory allocation.
- Memory Leak: Memory not released after use.
- Garbage Collection: Automatic memory cleanup (e.g., in Java, Python).

5. Paging & Virtual Memory

- Paging: Splits memory into fixed-size pages.
- Page Fault: Accessing data not currently in RAM.
- TLB: Caches page table mappings.
- Swapping: OS moves data between RAM and disk when needed.

6. Context Switching

- CPU switches between processes/threads.
- Saves and restores state of processes.
- Expensive operation; too many switches can degrade performance.