Lesson 5: Operating Systems

1. Introduction to the OS

Definition & Goals

- **Definition:** System software that acts as an intermediary between the user/applications and the computer hardware.
- **Main Goals:**
  - Provide a **Virtual Machine** (hides hardware complexity).
  - **Manage Resources** (CPU, memory, devices).
  - Provide an environment to **Execute Applications**.

Main Functions

- **Process Management:** Handling the execution, scheduling, and termination of processes.
- **Resource Management:** Allocating CPU time, memory, and I/O devices among competing processes.
- **Providing Interfaces:** Offers User Interfaces (GUI, CLI) for users and Application Program Interfaces (APIs) for software.
- **Security & Protection:** Controlling access to system resources and protecting processes from each other.

OS Classification

- **By Tasking:** Single-user/Single-task, Single-user/Multi-task, Multi-user/Multi-task.
- **Multi-threading:** Allows a single process to have multiple concurrent execution threads, improving performance.
- **Real-Time (RTOS):** For systems needing precise timing and high reliability (e.g., industrial control, medical devices).
- **Time-Sharing:** Shares CPU time among multiple users/tasks, providing interactivity through context switching.

2. Memory & Device Management

Memory Management

- **Virtual Memory:** Using secondary storage (disk) as an extension of RAM. Allows programs to be larger than physical memory.
- **Paging:** Dividing logical memory into fixed-size **pages** and physical memory into **frames** of the same size. Can cause *Internal Fragmentation* (wasted space within a frame).
- **MMU (Memory Management Unit):** The hardware component that translates virtual addresses generated by the CPU into physical addresses in main memory.

Device Management

- **Device Driver:** A specific software program that acts as a translator, allowing the OS to communicate with and control a hardware device.
- **Spooling (Simultaneous Peripheral...):** Using a buffer (usually on disk) to hold jobs for a slow device like a printer. This frees up the CPU to perform other tasks instead of waiting for the slow device to finish.

3. Process & Scheduling Management

Processes

- **Process vs. Program:** A `Program` is passive code on disk; a `Process` is a "program in execution" with its own resources.
- **Process States & Transitions:**
  - **New:** The process is being created.
  - **Ready:** Waiting to be assigned to a processor.
  - **Running:** Instructions are being executed.
  - **Waiting/Blocked:** Waiting for some event to occur (e.g., I/O completion).
  - **Terminated:** The process has finished execution.
- **PCB (Process Control Block):** A data structure holding all info about a process (State, ID, Program Counter, CPU Registers, Memory Limits, etc).
- **Context Switching:** The mechanism of saving one process's state (to its PCB) and loading another's to share the CPU. This process introduces overhead.
- **Interrupt:** A signal to the CPU indicating an event that needs immediate attention, pausing the current process.

Schedulers & Policies

Scheduler Comparison

- **Long-term (Job):** Admits jobs into the system. Controls degree of multiprogramming.
- **Short-term (CPU):** Selects the next process for the CPU. Fastest.
- **Medium-term (Swapper):** Swaps processes between memory and disk (manages suspended processes).

- **Policies:** *Non-preemptive* (process runs until it stops itself) vs. *Preemptive* (OS can force a process to stop).
- **Performance Metrics:**
  - **Turnaround Time:** Total time from submission to completion.
  - **Response Time:** Time from command to first response.
  - **Throughput:** Number of processes completed per unit time.
  - **Waiting Time:** Total time a process spends in the ready queue.

4. File & Storage Management

Files & Directories

- **File Attributes:** Name, Type (.exe, .txt), Size, Location, Owner, Access Permissions (Read/Write/Execute), Timestamps.
- **File Systems:** Defines how data is stored/retrieved.
  - **FAT:** Older, simple file system.
  - **NTFS:** Modern Windows standard. Better security, error recovery, large disk support.
- **File Security:** Managed via **Passwords** and **Access Privileges** (permissions) to control who can read, write, or execute files.

Storage Allocation Methods

Allocation Method Comparison

- **Contiguous:** File occupies one single block of disk space.
  - *Pro: Fast access. Con: External Fragmentation.*
- **Linked:** Blocks are scattered; each block points to the next.
  - *Pro: No external fragmentation. Con: Slow random access.*
- **Indexed:** An "index block" contains pointers to all other data blocks.
  - *Pro: Fast random access, no external fragmentation.*

- **Defragmentation:** Rearranging fragmented files into continuous blocks to improve disk performance.