Operating Systems Notes

Unit 1: Introduction and OS Structures

Core Concept: The OS is system software that acts as a manager between the user/hardware, making the system convenient to use and efficient.

- **OS Overview & Functions:**
  - Manages hardware (CPU, Memory, Disk, I/O devices).
  - Provides a user interface (CLI, GUI).
  - Executes and provides services for application software.
- **History & Types:**
  - **Batch:** Similar jobs grouped and executed together.
  - **Multiprogrammed:** Multiple jobs in memory; CPU switches when one is waiting for I/O.
  - **Time-Sharing/Multitasking:** Extension of multiprogramming with quick user response time.
  - **Real-Time:** For time-critical applications (e.g., Robotics, Flight Control). *Hard* (strict deadline) vs *Soft* (tolerable delays).
  - **Distributed:** Manages a group of independent computers as a single system.
- **OS Services:** User Interface, Program Execution, I/O Operations, File System Manipulation, Communication, Error Detection, Resource Allocation.
- **System Calls:** The programming interface for services provided by the OS. (e.g., read(), write(), fork()).
- **OS Architecture:**
  - **Monolithic:** Entire OS as a single program in kernel space. (Simple but less secure).
  - **Layered:** OS in hierarchical layers, each using services of the layer below.
  - **Microkernel:** Minimal kernel (IPC, basic scheduling). Other services run as user processes. (More stable, secure).

Video Links:

- **English:** Neso Academy - OS Introduction Playlist
- **Hindi: Gate Smashers - OS Introduction**
  - What is Operating System?
  - OS Structures & System Calls

Unit 2: Process and CPU Management

Core Concept: Managing the execution of processes to maximize CPU utilization and ensure fair processing.

- **Process:** A program in execution. More than just code (Program Counter, Stack, Data section).
- **Process Control Block (PCB):** Data structure holding all info about a process (PID, state, priority, registers, etc.).
- **Process States:** New, Ready, Running, Waiting, Terminated.

- **Context Switching:** The mechanism of saving the state of a running process and loading the state of the next process. It's pure overhead.
- **Scheduling Queues:** Job Queue, Ready Queue, Device Queues.
- **Threads:** A lightweight process (LWP). A basic unit of CPU utilization. Threads of a process share code, data, and files.
- **CPU Scheduling Algorithms:**
  - **FCFS (First-Come, First-Served):** Non-preemptive. Simple, can lead to convoy effect.
  - **SJF (Shortest Job First):** Optimal for average waiting time, but difficult to implement.
  - **Priority Scheduling:** Can be preemptive or non-preemptive. Problem: Starvation (low-priority jobs may never run).
  - **RR (Round Robin):** Preemptive FCFS with a time quantum (time slice). Good for time-sharing systems.

Video Links:

- **English:** Abdul Bari - Process Scheduling Algorithms
- **Hindi: Gate Smashers - CPU Scheduling**
  - Process States & PCB
  - Scheduling Algorithms

Unit 3: Concurrency and Synchronization

Core Concept: Managing multiple processes/threads that access shared data to prevent inconsistent results and handle deadlocks.

- **Critical Section Problem:** A section of code where a process accesses shared resources. The solution must ensure **Mutual Exclusion**, **Progress**, and **Bounded Waiting**.
- **Semaphores:** An integer variable (S) that can only be accessed via two atomic operations: wait(S) and signal(S). Used for synchronization.
  - **Binary Semaphore (Mutex):** Value 0 or 1.
  - **Counting Semaphore:** Unrestricted domain.
- **Monitors:** A high-level synchronization construct that allows only one process to be active inside it at a time. Safer and easier than semaphores.
- **Classical Problems:**
  - **Producer-Consumer:** Bounded-Buffer problem.
  - **Reader-Writer:** Multiple readers can read, but only one writer can write.
  - **Dining Philosophers:** Demonstrates deadlock and synchronization.
- **Deadlocks:** A situation where a set of processes are blocked because each is holding a resource and waiting for another resource acquired by some other process.
  - **Necessary Conditions:** Mutual Exclusion, Hold and Wait, No Preemption, Circular Wait.
  - **Prevention:** Break one of the four conditions.
  - **Avoidance (Banker's Algorithm):** The OS checks if granting a request will lead to a safe state.

    - **Detection & Recovery:** Allow deadlock to occur, then detect it and recover by aborting processes or preempting resources.

Video Links:

- **English:** Neso Academy - Process Synchronization
- **Hindi: Gate Smashers - Synchronization & Deadlock**
    - Semaphores & Mutex
    - Deadlock & Banker's Algorithm

Unit 4: Memory Management

Core Concept: Efficiently managing the primary memory to accommodate multiple processes, providing the abstraction of virtual memory.

- **Contiguous Allocation:** Each process is contained in a single contiguous section of memory.
    - **Fixed/Multiprogramming with fixed tasks (MFT):** Memory is divided into fixed-sized partitions.
    - **Variable/Multiprogramming with variable tasks (MVT):** Memory is divided into variable-sized partitions. Suffers from **External Fragmentation**.
- **Paging:** Solves external fragmentation. Physical memory is divided into fixed-sized blocks called **frames**. Logical memory is divided into same-sized blocks called **pages**. A **Page Table** is used to translate logical addresses to physical addresses.
- **Segmentation:** Memory-management scheme that supports the user's view of memory. A program is a collection of segments (e.g., code, stack, data). Can suffer from **External Fragmentation**.
- **Virtual Memory:** Technique that allows the execution of processes that are not completely in memory. Allows for larger logical memory than physical memory.
- **Demand Paging:** Pages are loaded only when they are demanded during program execution.
- **Page Replacement:** When a page fault occurs and no free frame is found, a page must be replaced.
    - **FIFO (First-In-First-Out):** Replace the oldest page.
    - **Optimal (OPT):** Replace the page that will not be used for the longest period. (Theoretical, used for comparison).
    - **LRU (Least Recently Used):** Replace the page that has not been used for the longest time.
- **Thrashing:** A state where the OS is spending more time paging (swapping pages in and out) than executing processes.

Video Links:

- **English:** Neso Academy - Memory Management
- **Hindi: Gate Smashers - Memory Management**
    - Paging & Segmentation
    - Page Replacement Algorithms

Unit 5: File and Storage Management

Core Concept: Providing a uniform logical view of information storage (files) and managing them efficiently on physical storage (disks).

- **File Concepts:** A named collection of related information. Has attributes like name, location, size, type.
- **File Access Methods:**
    - **Sequential:** Access records in order.
    - **Direct (Random):** Access records in any order.
- **Directory Structure:** Organizes files. Types: Single-Level, Two-Level, Tree-Structured, Acyclic-Graph.
- **File System Implementation:**
    - **Allocation Methods:** How disk blocks are allocated for a file.
        - **Contiguous:** File occupies contiguous disk blocks. Fast but suffers from external fragmentation.
        - **Linked:** Each block contains a pointer to the next block. No external fragmentation, but slow for direct access.
        - **Indexed:** All pointers are kept in a separate index block.
- **Free Space Management:** Tracking free disk blocks. Methods: **Bit Vector (Bitmap)**, **Linked List**.
- **Disk Scheduling Algorithms:** To minimize seek time (time for disk arm to move to the required cylinder).
    - **FCFS:** Simple, fair.
    - **SSTF (Shortest Seek Time First):** Services the request closest to the current head position. Can cause starvation.
    - **SCAN (Elevator Algorithm):** Moves the disk arm from one end to the other, servicing requests.
    - **C-SCAN:** A variant of SCAN that provides a more uniform wait time.

Video Links:

- **English:** Neso Academy - File Systems
- **Hindi: Gate Smashers - File & Disk Management**
    - File Allocation Methods
    - Disk Scheduling Algorithms

Unit 6: Device, Protection, and Security

Core Concept: Managing I/O devices and ensuring the system is protected from unauthorized access and malicious software.

- **I/O Systems:** The OS must manage communication with a wide variety of I/O devices.
- **Device Drivers:** Software modules that act as a translator between a hardware device and the OS/applications that use it.
- **Kernel I/O Subsystem:** Provides services like scheduling, buffering, caching, and spooling for I/O.

- **Protection:** Mechanism for controlling the access of processes and users to resources.
    - **Domain of Protection:** Users, Processes, Objects (files, memory segments).
    - **Access Matrix:** A model of protection. Rows represent domains, columns represent objects.
- **Security:** Defense of the system against internal and external attacks.
    - **Threats:** Worms, Viruses, Denial-of-Service (DoS).
    - **Authentication:** Verifying the identity of a user (e.g., Passwords, Biometrics).
    - **Access Control:** Implementing security policies (e.g., Access Control Lists - ACLs).

Video Links:

- **English (I/O):** Neso Academy - I/O Systems
- **Hindi (Security): 5 Minutes Engineering - OS Security**
    - Protection & Security

Unit 7: Advanced Topics and Shell Programming

Core Concept: Exploring modern OS architectures and gaining practical scripting skills.

- **Distributed OS:** Manages a group of independent computers and makes them appear to be a single computer. Handles **load sharing** and **computation migration**.
- **Multiprocessor OS (SMP - Symmetric Multiprocessing):** Multiple processors share the same physical memory and are under the control of a single OS instance.
- **Virtual Machines:** A software that creates a virtual environment (guest OS) that behaves like a real computer. The software that creates VMs is called a **hypervisor** (e.g., VMware, VirtualBox).
- **Microkernels:** A minimal OS kernel that provides only essential services (IPC, basic scheduling, memory management). Other services (file system, device drivers) run as user-level servers. More reliable and secure.
- **Shell Programming (Scripting):** Writing scripts to automate tasks in the OS command-line interface (Shell).
    - **Variables, Loops (**for, while**), Conditionals (**if-else**).**
    - **Commands (**grep**,** awk**,** sed**).**
- **Performance Evaluation:** Measuring system performance (throughput, response time, CPU utilization).

Video Links:

- **English (Virtualization):** What is a Virtual Machine?
- **Hindi (Shell Scripting): Great Learning - Shell Scripting**
    - Shell Scripting Tutorial
- **Hindi (Advanced OS): University Academy - Advanced OS**
    - Distributed OS
    - Virtual Machines