Operating system

GATEWAY CLASSES

ONE SHOT REVISION

Unit-1

Introduction

CS / IT / CS Allied / MCA

By Dr. Kapil Kumar Sir

Gateway classes 7455 812 84

Operating System

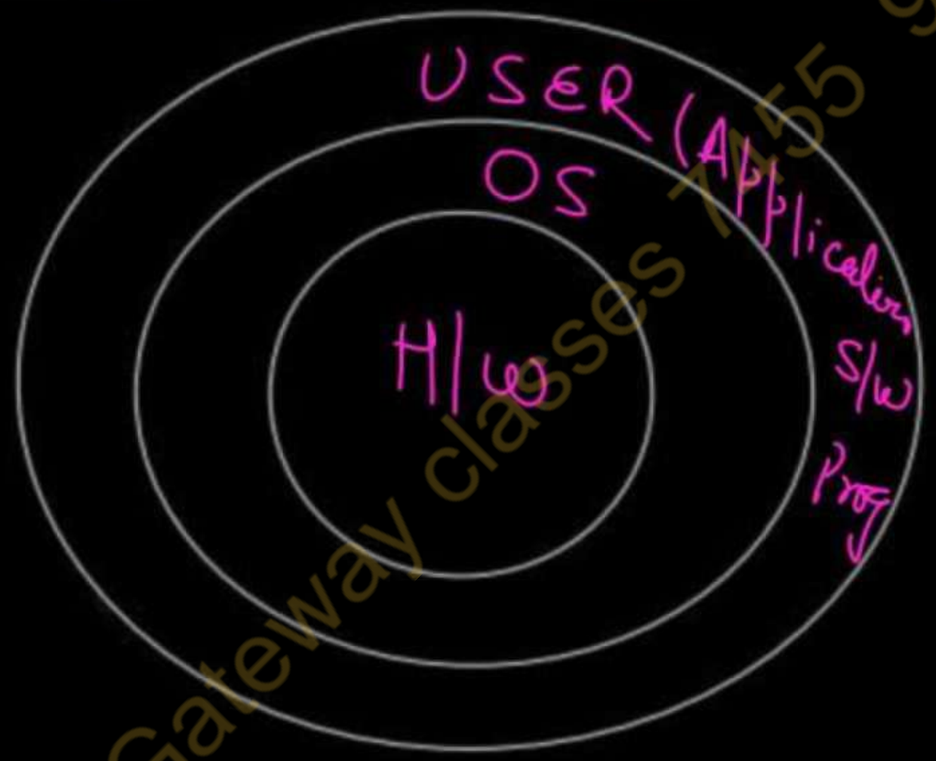
Syllabus – Unit - 1

Introduction: Operating system and functions, Classification of Operating systems- Batch, Interactive, Time sharing, Real-Time System, Multiprocessor Systems, Multiuser Systems, Multiprocess Systems, Multithreaded Systems, Operating System Structure- Layered structure, System Components, Operating System services, Reentrant Kernels, Monolithic and Microkernel Systems.

Gatewayclasses 7455 9612 84

What is an Operating System.....

- An operating system is a program that manages the computer hardware (H/W).
- An operating system provides a basis for application programs and acts as an intermediary between the user and the H/W.



3 ITEMS
3 ENTITIES

1. H/W – Machine
2. Application S/W
3. System S/W OS

S/W

Fig: - Position of Operating System

Gateway classes 7455 961284

- Operating system create an environment in which a user can easily interface with computer H/W to execute programs.

Components or Computer System:

- A computer system can be divided into four components.

(i) The H/W

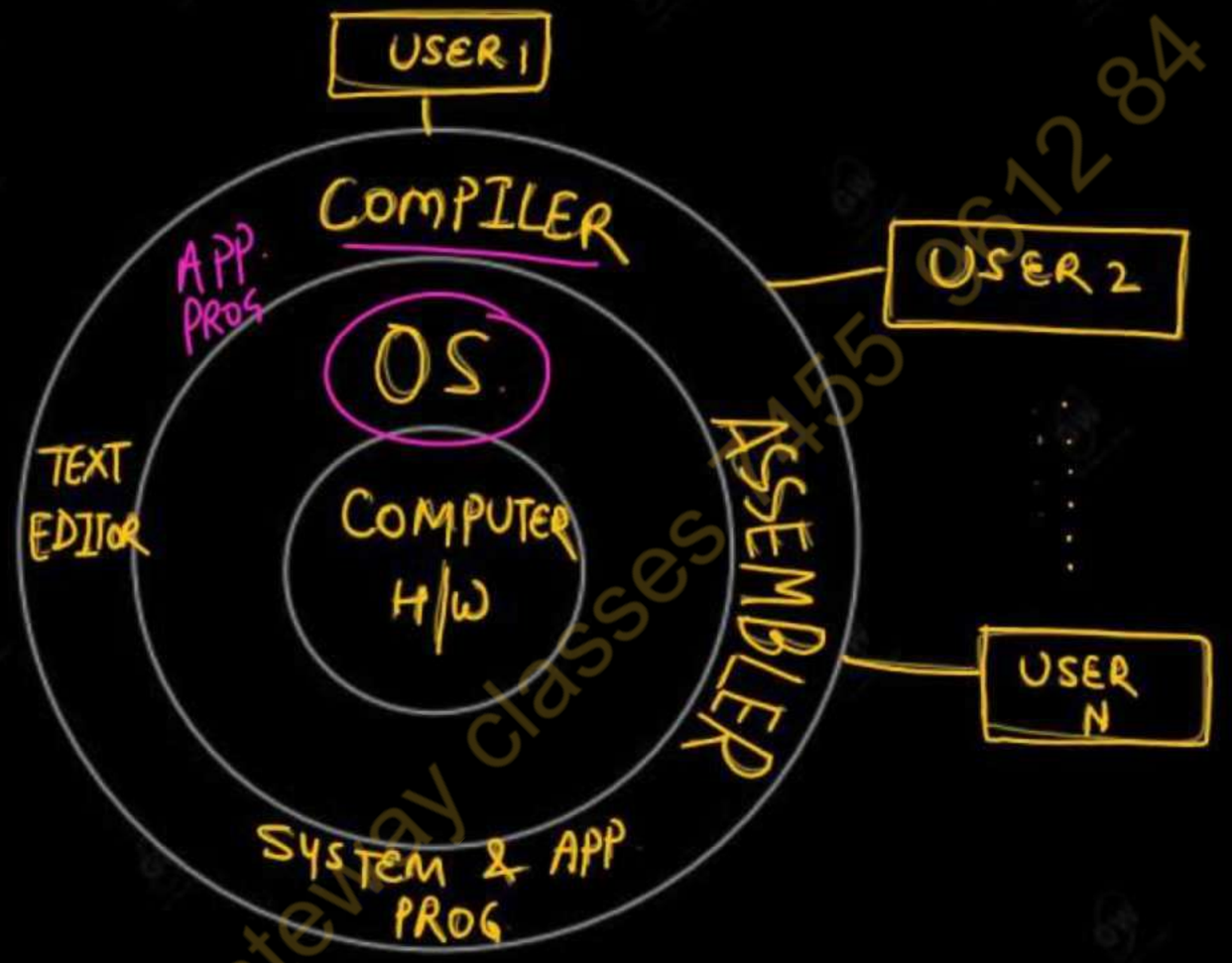
(ii) The O.S.

(iii) The Application programs &

(iv) The users.

Gateway classes 7455 9612 84

The abstract view of the components of a computer system can be presented in the following figure :



TRANSLATORS :-

1) ASSEMBLER

2) { COMPILER
INTERPRETER

HLL → 0,1

Fig: Abstract view of the components of a computer system.

Gateway classes 7455 96 12 84

The H/W may includes:-

- The memory
- Input & Output (I/O) devices
- CPU etc. which provides the basic computing resources for the system.

- The application programs such as word processors, spreadsheet packages, presentation packages, compilers, text editors, databases (DBs), web browsers, games, other applications, etc. define the ways in which these resources are used to solve user's computing problems.
- The Operating System controls the H/W and coordinates its use among the various application programs for the various users.
- The Operating System provides the means for the proper use of these resources in the operation of the computer system.

Gateway classes 7455 9612 84

User's and System's Point of View of the Operating System

To understand the operating system's role, we explore the OS from two viewpoints:

(i) User view and (ii) System view

Handheld - effiency PC - efficiency
Mainframe = accuracy Performance

- **User View:**

➢ From the user's perspective, the operating system provides a platform for executing applications and managing resources.

➢ Users interact with the operating system through graphical user interfaces (GUIs), command-line interfaces (CLIs), or application programming interfaces (APIs).

➢ User view focuses on tasks such as running applications, managing files, configuring settings, and interacting with peripherals.

➢ Users are concerned with the usability, responsiveness, and accessibility of the operating system's features and functions.

Gateway classes 7455 961284

- **System View**

Resource Manager

- The system view of the operating system deals with how the operating system manages hardware resources and provides services to applications.

Imp

- It includes components such as the kernel, device drivers, memory management, process management, file systems, and networking protocols.
- The system view emphasizes efficiency, resource allocation, security, and stability of the operating system.

a = b + c

- System administrators, developers, and software engineers often work with the system view to optimize performance, troubleshoot issues, and develop new system features.

Imp

- The user view focuses on interaction and usability for end-users, while the system view is related to the internal workings and management of resources within the operating system.

Gateway classes 7455 961284

Q :-

Kernel and its operations

- The kernel in an operating system is the core component responsible for managing system resources and providing essential services to other parts of the operating system and user applications.
- It acts as a bridge between hardware and software, enabling communication and coordination between different parts of the computer system.
- Key functions of the kernel include:

- **Process Management:** The kernel manages processes, allocating resources such as CPU time and memory to different tasks and ensuring proper execution.
- **Memory Management:** It oversees the allocation and deallocation of memory, handling memory protection, virtual memory, and memory mapping.
- **Device Management:** The kernel interacts with hardware devices, handling device drivers and providing an interface for applications to communicate with peripherals.

Gateway classes 7455 961284

- **File System Management:** It manages file operations, including reading, writing, and organizing data on storage devices.
- **Security and Access Control:** The kernel enforces security policies, controlling access to resources and ensuring the integrity and confidentiality of system operations.

Types of Kernel:-

1. **Monolithic Kernel** – All OS services run in the same (one) address space as the kernel itself.
2. **Microkernel** – Small and delegating most Operating System services to user space.
3. **Hybrid Kernel** – Combined aspects of both monolithic and microkernel.

Gateway Classes 7455 961284

Computer System Organization

Computer System Operation:-

A modern general purpose computer system consists of one or more CPUs and a number of device controllers connected through a common bus that provides access to shared memory.

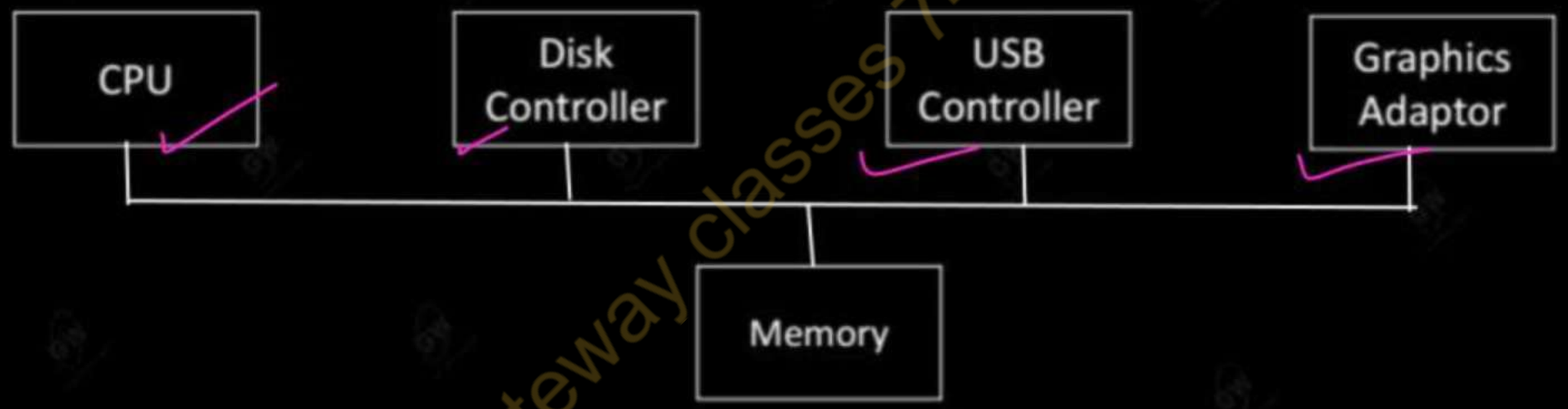


Figure: A Modern Computer System.

Gateway classes 7455 961284

- Each device controller is in charge of a specific type of device for example – disk drives audio devices and video displays.

{ init } Prog

- The CPU and device controllers can execute concurrently, competing for memory cycles.
- To ensure orderly access to the shared memory a memory controller is provided whose function is to synchronize access to the memory.

Imp

- For a computer to start running – for instance when it is powered up or rebooted – it needs to have an initial program to run. This initial program or bootstrap program is stored in ROM (EPROM), known by the general term firmware within the computer H/W.

firmware

PROM EPROM

- It initializes all aspects of the system, from CPU registers to device controllers to memory contents.
- The bootstrap program must have the idea of how to load the operating system in the memory and how to start executing that system. To achieve this goal the bootstrap program must locate and load into memory the operating system.

Gateway classes 7455 962 284

What is a Bootstrap Program and How it Works?

- The bootstrap program, also known as the **bootstrap loader**, is a small piece of code that resides in the firmware or read-only memory of a computer system.
- Its primary purpose is to initialize the system hardware and load the operating system into memory to start the computer.
- It works in the following manner-

- **Power-On or Reset:** When the computer is powered on or restarted, the processor executes a predefined start-up routine that begins fetching instructions from a predetermined location in the system firmware.
- **Bootstrap Program Execution:** The processor executes the instructions stored in the firmware's bootstrap program. The bootstrap program performs basic hardware initialization tasks, such as checking the integrity of the system hardware and initializing essential components like memory, CPU registers, and peripheral devices.

Gatewayclasses 7455 9612 84

- **Loading the Operating System:** After completing the hardware initialization, the bootstrap program locates and loads the operating system's kernel into the system's main memory (RAM) from a boot device, such as a hard disk drive.
- This process involves reading the initial boot sector or boot block from the designated boot device, which contains the necessary instructions to load the rest of the operating system kernel into memory.
- **Handoff to the Operating System:** Once the operating system kernel is successfully loaded into memory, control is transferred to the kernel, and the bootstrap program's execution is completed.
- The operating system takes over and continues the bootstrapping process, initializing its components, establishing system configurations, and transitioning to the user mode to start executing user applications.

Gateway classes 7455 9612 84

- User mode and kernel mode represent different privilege levels in the operating system, with user mode providing restricted access to system resources for user processes, while kernel mode grants full control and access to the operating system kernel to manage system resources and perform privileged operations.

Gateway classes 7455 961284

Functions of Operating System

1. Memory Management

- The operating system manages the Primary Memory or Main Memory.
- Main memory is made up of a large array of bytes or words where each byte or word is assigned a certain address.
- Main memory is fast storage and it can be accessed directly by the CPU.
- For a program to be executed, it should be first loaded in the main memory.

An Operating System performs the following activities for Memory Management:

- It keeps track of primary memory, i.e., which bytes of memory are used by which user program.
- The memory addresses that have already been allocated and the memory addresses of the memory that has not yet been used.
- In multiprogramming, the OS decides the order in which processes are granted memory access, and for how long.
- It allocates the memory to a process when the process requests it and deallocates the memory when the process has terminated or is performing an I/O operation.

Gateway classes 7455 961284

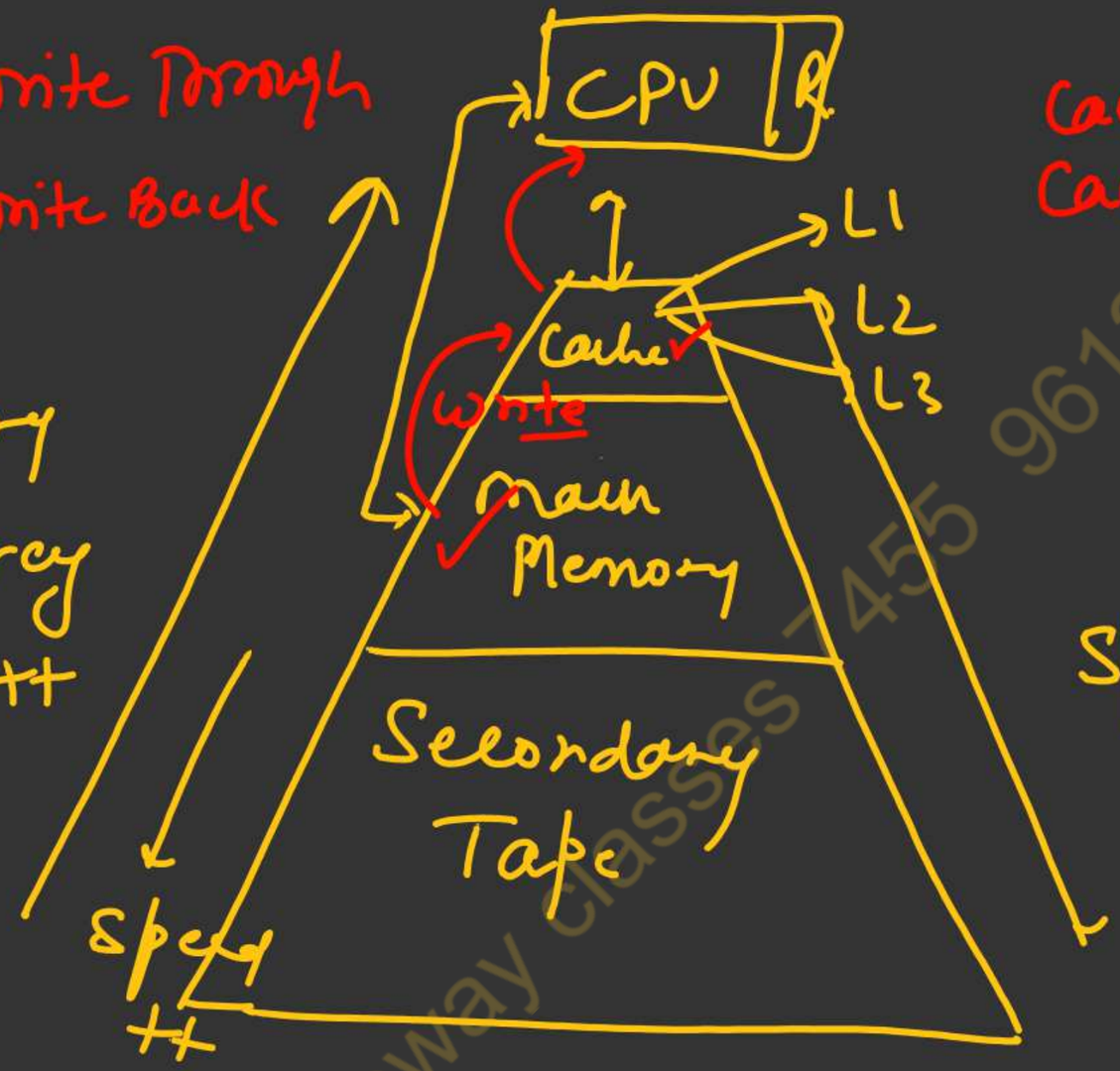
2. Caching

- Information is normally kept in some storage system such as main memory.
- As it is used, it is copied into faster storage system the cache on a temporary basis.
- The operating system plays a crucial role in caching, which involves storing frequently accessed data in a high-speed storage location (cache) to reduce access latency and improve system performance.
- Here's how the operating system contributes to caching:

- **Managing Memory Hierarchy:** The operating system organizes the computer's memory hierarchy, which typically includes multiple levels of cache (L1, L2, L3), main memory (RAM), and secondary storage (e.g., hard drives, SSDs).
- **Controlling Cache Policies:** The operating system implements cache replacement policies to determine which data to keep in the cache when new data needs to be stored. Common cache replacement algorithms include Least Recently Used (LRU), First-In-First-Out (FIFO), and Random Replacement. The choice of policy affects cache performance and efficiency.

Gateway classes 7455 96 12 04

Cache W → Write Through
→ Write Back



Memory Hierarcy

speed ++

speed ++

Cache hit

Cache miss

space--

space++

```
for(i=0; i<10; i++)
sum=sum+a[0];
```

Gateway classes 7455 9612 84

- **Managing Cache Coherency:** In multi-core or distributed systems where multiple processors or devices share access to cached data, the operating system ensures cache coherency to maintain consistency across caches. It coordinates cache invalidation, updates, and synchronization to prevent data inconsistencies and ensure correct program execution.

➢ [Write through and write back]

➢ When we need a particular piece of information, we first check whether it is in the cache. If it is, we use the information directly from the cache; if it is not, we use the information from the source, putting a copy in the cache under the assumption that we will need it again soon. Because the cache has a limited size, "cache management" is an important design problem.

➢ Careful selection of cache size and replacement policy can result in greatly increased performance.

Gateway classes 7455 961284

3. Processor Management

- The operating system manages processes, which are programs in execution.
- It allocates system resources, such as CPU time and memory space, to processes and ensures their efficient execution.
- In a multi-programming environment, the OS decides the order in which processes have access to the processor, and how much processing time each process has.
- This function of OS is called Process Scheduling.
- An Operating System performs the following activities for Processor Management.

- It keeps track of the status of processes.
- Allocates the CPU that is a processor to a process.
- De-allocates processor when a process is no more required.

Gateway classes 7435 961284

4. Device Management

- Operating systems control and coordinate peripheral devices, such as printers, keyboards, and disk drives, by managing device drivers and providing a uniform interface for device interaction.
- An OS performs the following activities for device management.

- Keeps track of all devices connected to the system.
- Designates a program responsible for every device known as the Input/output controller. Decides which process gets access to a certain device and for how long.
- Allocates devices effectively and efficiently.
- Deallocates devices when they are no longer required.

Gateway classes 7455 961284

5. File System Management

- The operating system provides a file system that organizes and manages files stored on secondary storage devices, such as hard drives or SSDs.
- It handles file creation, deletion, modification, and access, ensuring data integrity and security.
- A file system is organized into directories for efficient or easy navigation and usage.
- These directories may contain other directories and other files.
- An Operating System carries out the following file management activities.
- It keeps track of where information is stored, user access settings, the status of every file, etc.
- These facilities are collectively known as the file system.

Gateway classes 7455 9612 84

6. User Interface or Command Interpreter

- The user interacts with the computer system through the operating system.
- Hence OS act as an interface between the user and the computer hardware.
- This includes command-line interfaces (CLI), graphical user interfaces (GUI), or even touch-based interfaces for mobile devices, facilitating user interaction and system control.
- Through this interface, the user makes interaction with the applications and the machine hardware.

DOS-CLI
GUI - Window
Touch

7. Booting the Computer

- The process of starting or restarting the computer is known as booting.
- It is of two types:

a. Cold booting

b. Warm booting

- If the computer is switched off completely and if turned on then it is called cold booting.
- Warm booting is a process of using the operating system to restart the computer.

Gateway classes 7455 96 12 34

8. Protection & Security

- Operating systems enforce security policies to protect system resources and data from unauthorized access, viruses, malware, and other threats.
- They manage user authentication, permissions, and encryption to ensure system integrity and confidentiality.
- The operating system uses password protection to protect user data and similar other techniques.
- It also prevents unauthorized access to programs and user data.

9. Control Over System Performance

- The operating system monitors overall system health to help improve the overall performance of the system.
- Records the response time between service requests and system response to have a complete view of the system's health.
- This can help improve performance by providing important information needed to troubleshoot problems.

Gateway classes 7455 96 12 84

10. Job Accounting

- The operating system Keeps track of time and resources used by various tasks and users, this information can be used to track resource usage for a particular user or group of users.
- Job accounting refers to the process of tracking resource usage and activities associated with individual processes or jobs running on a computer system.
- The operating system plays several key roles in job accounting:
- **Resource Monitoring:** The operating system monitors resource usage by processes, including CPU time, memory usage, disk I/O, and network activity. It tracks these metrics to provide insights into the performance and efficiency of individual jobs.
- **Data Collection:** The operating system collects data on resource consumption and job attributes, such as process identifiers (PIDs), start and end times, user IDs, and execution priorities. This information is logged and stored for later analysis and reporting.

Gatewayclasses 7455 961284

- **Accounting Records:** The operating system maintains accounting records for each executed job, recording relevant information such as resource usage, job identifiers, and user IDs. These records serve as a historical log of system activity and are used for billing, auditing, and performance analysis purposes.
- **Billing and Cost Allocation:** In multi-user or multi-tenant environments, the operating system supports billing and cost allocation by tracking resource usage for each user or account. It calculates usage-based charges or quotas based on predefined billing policies and accounting data.
- **Auditing and Compliance:** The operating system facilitates auditing and compliance requirements by generating audit trails and reports based on job accounting data. It ensures accountability and transparency by documenting system activities and user actions for security and regulatory purposes.

Gateway classes 7455 961284

- **Performance Analysis:** Job accounting data collected by the operating system is used for performance analysis and optimization. System administrators and performance analysts analyze resource usage patterns, identify bottlenecks, and optimize system configurations to improve overall efficiency and performance.
- **Policy Enforcement:** The operating system enforces resource usage policies and quotas based on job accounting data. It may impose limits on CPU time, memory usage, or disk space to prevent resource contention, prioritize critical workloads, and maintain system stability.

11. Error-detection and Correction:

Debugging

- Operating systems detect, report, and recover from errors, faults, and exceptions occurring within the system.
- They provide mechanisms for error logging, debugging, and fault tolerance to maintain system stability and reliability.

Gateway classes 7455 961284

12. Coordination between other software and users

- Operating systems also coordinate and assign interpreters, compilers, assemblers, and other software to the various users of the computer systems.

13. Network Security and Management

- The OS provides network connectivity and manages communication between computers on a network.
- It also manages network security by providing firewalls (software or hardware based security system) and other security measures.
- Modern operating systems include networking capabilities to enable communication and data exchange between computers and devices over networks. They support network protocols, configurations, and services for seamless connectivity.

Gateway classes 7455 9612 82

AKTU PYQs

Q.1. What do you mean by kernel? 2015-16, 2 Marks

Q.2. Define the Operating system explain in short. 2016-17, 2 Marks

Q.3. What is a kernel? Describe various operations performed by the kernel. 2016-17, 7 Marks

Q.4. Define the operating system. List the objectives of an operating system. 2017-18, 2 Marks

Q.5. What is an operating system? 2017-18, 2 Marks

Q.6. What is an operating system? Describe the operating system functions. 2018-19, 7 Marks

Q.7. Define kernel and names of types of kernel. 2018-19, 2 Marks

Q.8. Define the operating system and mention its major functions. 2021-22, 2 Marks

Q.9. Explain two main functions of an operating system. 2022-23, 2 Marks

Gateway classes 7455 9612 84

Classification of Operating System

1. Batch Operating System
2. Timesharing or Fair Share or Multitasking Operating System
3. Real Time Operating System (RTOS) (Hard RTOS and Soft RTOS)
4. Multiprocessor Operating System (Symmetric and Asymmetric)
5. Multiuser Operating System
6. Single Threaded & Multithreaded Operating System
7. Multiprogramming Operating System

Gateway classes 7455 9612 84

1. Batch Operating System:-

A Batch operating system is a type of operating system where tasks or jobs are grouped together and executed sequentially without manual intervention.

- In a batch processing environment, users submit their jobs to the operator then the operator organizes these jobs into batches and provide to the system to processes them one after another.
- In Batch Operating system, the "Operator" takes similar jobs having the same requirement and groups them into batches. It is the responsibility of the operator to select jobs with similar needs.
- Key characteristics of a Batch operating system include:

- **No User Interaction During Execution:** Once a batch of jobs is submitted, the system executes them automatically without requiring further input from the user.
- **Efficient Resource Utilization:** Batch processing allows the operating system to efficiently utilize computing resources by executing multiple jobs in sequence without leaving the CPU idle.

Gatewayclasses 7455 9612 84

JOB

I — COBOL = 5

FORTRAN = 2

C = 5

⋮

COBOL

F

C

Gateway Classes 7455 9612 84

- **Limited Interactivity:** Since users submit their jobs as batches, there is typically limited interaction with the system during job execution. Users may receive notification of job completion or errors after the batch has been processed.
- **Job Scheduling:** The operating system's scheduler determines the order in which jobs within a batch are executed based on priority, resource availability, and other factors.
- **Commonly Used in Background Processing:** Batch operating systems are commonly used for tasks that don't require immediate user interaction, such as running large-scale computations, generating reports, or processing data in the background.
- Example :- OS/360

Applications Areas:

- ❑ Batch operating systems are designed to efficiently process large volumes of work in a sequential manner, making them suitable for various types of applications where immediate user interaction is not necessary.

Gateway classes 7455 961284

Disadvantages:

- **Limited Interactivity:** Batch operating systems lack real-time interaction with users during job execution. This can be a drawback for applications requiring immediate feedback or user interaction, as users have to wait for the completion of batch processing to receive results or provide input.

  B = 5 ← 2, 10, 3, 2 → 18

- **Increased Turnaround Time:** Since jobs are processed sequentially in batches, there may be delays in starting the execution of a particular job, especially if the system is processing a large backlog of jobs. This increased turnaround time can affect the responsiveness of the system, particularly for time-sensitive tasks.
- **Difficulty in Debugging:** Debugging and troubleshooting batch processing jobs can be challenging, especially when errors occur during execution. Identifying the cause of errors and diagnosing problems may require reviewing logs or analyzing job outputs after completion, which can be time-consuming and complex.

Gateway classes 7455 96 12 84

- **Increased Overhead:** Batch processing often involves additional overhead in job submission, scheduling, and management compared to interactive processing systems. This overhead can consume system resources and may impact overall system performance, particularly during peak workload periods.
- **Risk of Job Failure:** In batch operating systems, the failure of one job within a batch can potentially affect the execution of subsequent jobs in the same batch. This risk of cascading failures can impact the reliability and robustness of the system, particularly if error recovery mechanisms are not robustly implemented.

Gateway classes 7455 9612 64

2. Time Sharing or Multitasking or Fair Share Operating System :

- A time-sharing operating system is a type of operating system that allows multiple users to share a single computer simultaneously.
- In a time-sharing system, the CPU time is divided into small time slices or intervals, and each user or task is allocated a portion of this time slice to execute their program or task.
- Examples : Unix, Linux, Windows, Solaris, OpenVMS etc.
- Key features and characteristics of Time Sharing operating systems include:

- **Multiprogramming:** Time-sharing systems support multiprogramming, allowing multiple programs to run concurrently. Each program is allocated a small portion of CPU time, and the CPU switches rapidly between programs to give the illusion of simultaneous execution.
- **Interactive Environment:** Time-sharing systems provide an interactive computing environment, allowing users to interact with the system in real-time through terminals or other input/output devices. Users can run programs, execute commands, and receive immediate feedback from the system.

Gateway classes 7465 96 12 84

- **Fair Resource Allocation:** Time-sharing systems typically use scheduling algorithms to allocate CPU time fairly among active processes or users. These algorithms ensure that each user or task receives a fair share of CPU time, preventing any single user from monopolizing the system resources.
- **Concurrent Access:** Time-sharing systems support concurrent access to system resources such as memory, disk, and peripherals. Multiple users can access and manipulate shared resources simultaneously without interfering with each other's operations.
- **Context Switching:** Time-sharing systems rely on context switching to rapidly switch between different programs or tasks. When a time slice expires or when a process needs to wait for input/output operations, the system saves the current state of the process (context) and switches to another process.

Gateway classes 7455 961284

Advantages:

- **Resource Sharing:** Time-sharing systems allow multiple users to share a single computer system simultaneously. This enables efficient utilization of computing resources, as the CPU, memory, and other system resources are shared among multiple users and tasks.
- **Cost-Efficiency:** By allowing multiple users to share a single computer system, time-sharing operating systems can reduce hardware and infrastructure costs.
- **Increased Utilization:** Time-sharing systems maximize the utilization of computing resources by multiplexing CPU time among multiple users and tasks. This helps prevent idle time and ensures that the CPU is continuously engaged in processing tasks, improving overall system efficiency.

Gateway Classes 7455 961284

- **Interactive Environment:** Time-sharing systems provide an interactive computing environment, allowing users to interact with the system in real-time through terminals or other input/output devices. Users can run programs, execute commands, and receive immediate feedback from the system, enhancing productivity and user experience.
- **Fairness and Equity:** Time-sharing systems use scheduling algorithms to allocate CPU time fairly among active processes or users. These algorithms ensure that each user or task receives a fair share of CPU time, preventing any single user from monopolizing the system resources and promoting equity among users.
- **Improved Turnaround Time:** Time-sharing systems reduce the turnaround time for user tasks by providing rapid response and execution of commands and programs. Users can initiate tasks and receive results promptly, leading to improved productivity and efficiency.

Gateway classes 7455 9612 84

Disadvantages:

- **Overhead from Context Switching:** Time-sharing systems incur overhead from frequent context switching between different processes or users.
- **Complexity of Scheduling Algorithms:** Scheduling algorithms used in time-sharing systems must balance fairness, responsiveness, and resource utilization. Designing and implementing efficient scheduling algorithms can be complex. RR
- **Security Risks:** Time-sharing systems pose security risks due to the shared nature of resources among multiple users. Users may inadvertently or maliciously access or modify each other's data, compromising system security and confidentiality.
- **Risk of Resource Contention:** In time-sharing systems, multiple users compete for access to shared system resources such as CPU time, memory, and I/O devices. Resource contention can occur, leading to delays in task execution and decreased system responsiveness.

Gateway classes 7455 96 12 84

3. Real Time Operating System

RTOS

- A real-time operating system (RTOS) is a specialized type of operating system designed to manage and execute tasks with precise timing constraints.
- RTOS focuses on deterministic behavior and meeting strict deadlines for task execution.
- Example : VxWorks, QNX, FreeRTOS,RTOS32 etc.

Key characteristics of real-time operating systems include:

- **Predictable Timing:** RTOS guarantees that tasks will be executed within specified time constraints, known as deadlines. This deterministic behavior ensures that critical tasks are completed on time, making RTOS suitable for applications where timing is crucial, such as industrial control systems, robotics, and embedded systems.
- **Task Scheduling:** RTOS employs scheduling algorithms optimized for real-time performance, such as fixed-priority scheduling or earliest deadline first (EDF) scheduling. These algorithms prioritize tasks based on their deadlines and criticality, ensuring that high-priority tasks are executed without delay.

Gateway classes 7455 96 12 84

- **Minimal Latency:** RTOS minimizes the latency, or delay, between the occurrence of an event and the execution of the corresponding task. Low latency is essential for time-sensitive applications where immediate response is required, such as in automotive systems or medical devices.

Types of RTOS: Hard Real-Time and Soft Real-Time:

- **Hard real-time systems** have strict timing requirements, where missing deadlines can lead to system failure or safety hazards.
- **Soft real-time systems** have more relaxed timing constraints, where occasional deadline misses are acceptable without catastrophic consequences.

Gateway classes 7455 96 12 84

Disadvantages of RTOS: -

- **Complexity and Development Cost:** Developing real-time systems with RTOS can be more complex and time-consuming compared to traditional systems. Real-time applications often require careful consideration of timing constraints, task scheduling, and resource management, leading to increased development costs and effort.
- **Limited Flexibility:** RTOS prioritizes deterministic behavior and meeting strict deadlines, which can limit the flexibility of the system. Changes to task priorities or system configurations may impact real-time performance and require careful validation and testing to ensure correctness.
- **Limited Scalability:** Real-time systems may face scalability limitations, particularly in terms of handling increasing task loads or adapting to changing system requirements. Scaling a real-time system to accommodate higher workloads may require careful redesign and optimization to maintain timing guarantees.

Gateway classes 7455 9612 84

- **Difficulty in Debugging:** Debugging real-time systems can be challenging due to the inherent timing constraints and non-deterministic behavior introduced by external factors. Identifying and diagnosing timing-related issues, such as missed deadlines or race conditions, may require specialized debugging techniques and tools.
- **Dependency on Hardware and Environment:** Real-time performance in RTOS can be influenced by factors such as hardware characteristics, system configuration, and environmental conditions. Variations in hardware timing behavior, system configuration changes, or external disturbances can impact the predictability and reliability of real-time systems.

Gateway classes 7455 961284

4. Multiprocessor Operating System :-

PE_0, PE_n CPU

- A multiprocessor operating system is an operating system designed to efficiently manage and coordinate multiple processors within a single computer system. ~~multicomputer~~
- Unlike single-processor systems where a single CPU executes tasks sequentially, multiprocessor systems have multiple CPUs that can execute tasks concurrently, leading to improved performance and scalability.
- Examples : Linux, Windows Server, MacOS, Solaris etc.

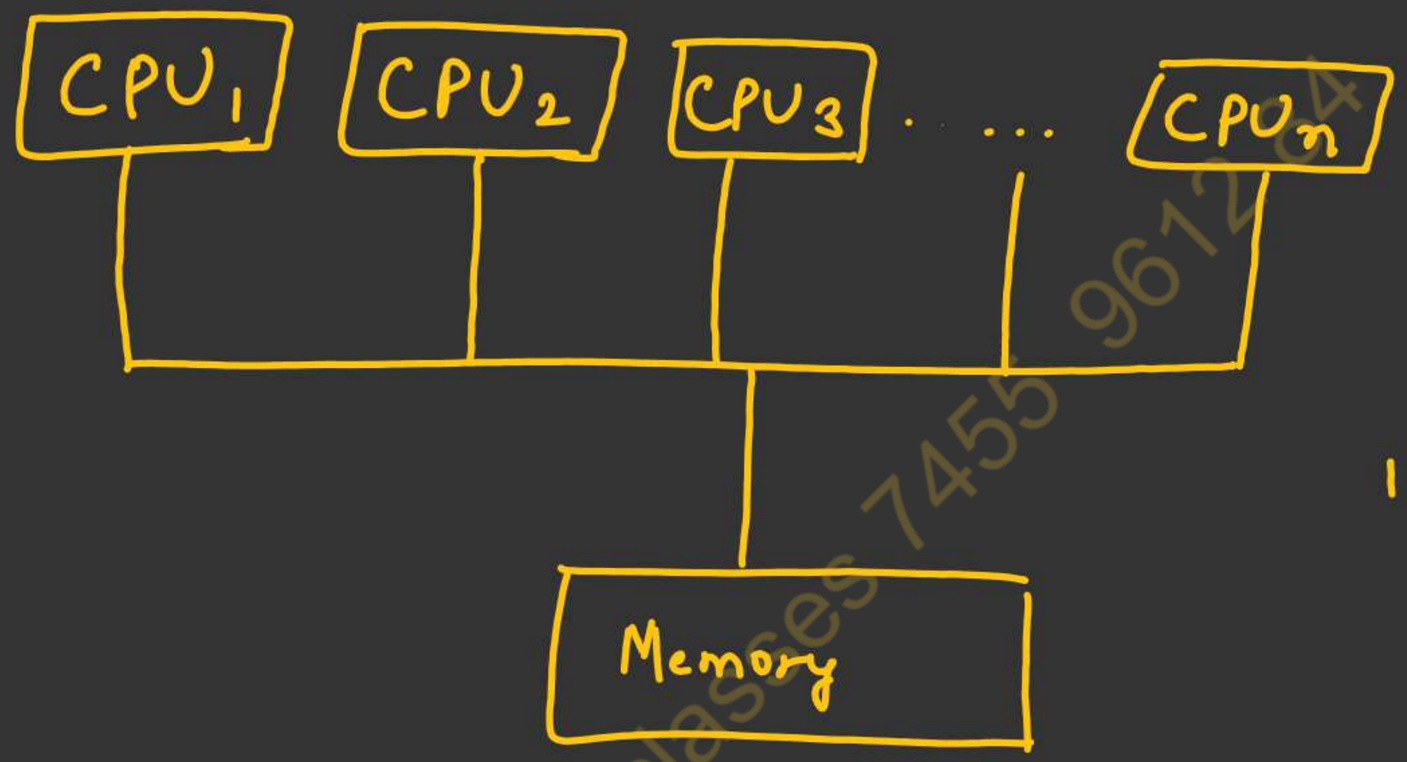
FLYNN'S CLASSIFICATION

1. SISD
2. SIMD
3. MISD
4. MIMD

Key characteristics of multiprocessor operating systems include:

- **Parallel Processing:** Multiprocessor operating systems support parallel processing, allowing multiple CPUs to execute tasks simultaneously. This parallelism enables faster execution of programs and better utilization of system resources, leading to improved system performance.
- **Task Distribution:** The operating system is responsible for distributing tasks among multiple CPUs to achieve load balancing and maximize CPU utilization.

Gatewayclasses 7455 96 12 84



1. Symmetric -

One OS Control all CPU

2. Asymmetric -

one master processor that give instruction to other CPU.

Gateway classes 7455 961264

- **Shared Memory:** Multiprocessor systems typically use shared memory architecture, where all CPUs have access to a common address space and can communicate with each other by reading from and writing to shared memory locations. Shared memory facilitates interprocessor communication and synchronization.
- **Interprocessor Communication:** Multiprocessor operating systems provide mechanisms for interprocessor communication, allowing CPUs to exchange data, coordinate activities, and synchronize execution. Examples of interprocessor communication mechanisms include message passing, shared memory, and synchronization primitives such as semaphores and locks.
- **Scalability:** Multiprocessor systems are highly scalable, allowing additional CPUs to be added to the system to handle increasing workloads or accommodate growing numbers of users. Scalability is achieved through efficient task distribution, load balancing, and resource management across multiple processors.

Gateway classes 7455 9612 84

- **Fault Tolerance:** Some multiprocessor operating systems include features for fault tolerance and redundancy, such as redundant CPUs or failover mechanisms. These features help ensure system reliability and availability by tolerating hardware failures or errors.
- These systems are widely used in modern computing environments, ranging from servers and workstations to high-performance computing clusters and supercomputers, where increased computational power and throughput are essential.

Gateway classes 7455 961 264

Disadvantages:

- **Complexity:** Multiprocessor systems are inherently more complex than single-processor systems due to the need for coordinating and managing multiple CPUs.
- **Synchronization Overhead:** Coordinating the activities of multiple CPUs in a multiprocessor system requires synchronization mechanisms to ensure data consistency and avoid race conditions.
- **Cache Coherence Issues:** Multiprocessor systems typically use shared memory architectures with caches at each CPU, leading to cache coherence issues.
- **Contention:** Multiprocessor systems may experience contention, particularly in shared resource scenarios where multiple CPUs compete for access to shared resources such as memory, I/O devices, or network interfaces. Contention can lead to performance degradation and reduced scalability, requiring careful resource management and optimization.

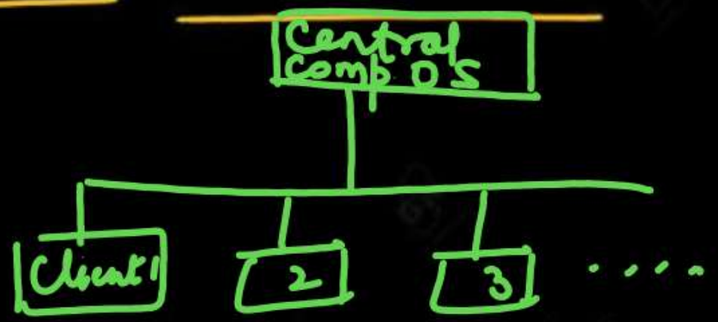
Gateway classes 7455 961 2 84

- **Complex Debugging and Testing:** Debugging and testing multiprocessor systems can be more challenging than single-processor systems due to the increased complexity and concurrency. Identifying and diagnosing concurrency-related bugs, race conditions, and synchronization issues may require specialized debugging techniques and tools, increasing development time and effort.
- **Higher Cost:** Multiprocessor systems typically incur higher hardware and software costs compared to single-processor systems due to the need for additional CPUs, memory, interconnects, and specialized hardware components. The cost of implementing and maintaining a multiprocessor system can be prohibitive for some applications or organizations, particularly those with budget constraints.

Gateway classes 7455 961284

5. Multiuser Operating Systems:

- Multiuser operating systems are designed to allow multiple users to access and interact with a computer system simultaneously.
- These operating systems provide mechanisms for managing user accounts, enforcing access control policies, and facilitating concurrent access to system resources.
- Examples : Unix, Linux, MAC OS, Solaris etc.



Some key features and characteristics of multiuser operating systems:

- **User Authentication:** Multiuser operating systems require users to authenticate themselves before accessing the system. Authentication mechanisms, such as usernames and passwords, are used to verify the identity of users and grant them appropriate access privileges.
- **User Account Management:** Multiuser operating systems support the creation, management, and deletion of user accounts. System administrators can create user accounts, assign passwords, and configure access permissions for individual users.

Gateway classes 7455 961284

- **Access Control:** Multiuser operating systems enforce access control policies to regulate user access to system resources. Access control mechanisms, such as file permissions and user groups, are used to restrict or permit access to files, directories, devices, and other system resources based on user identities and privileges.
- **Concurrent Access:** Multiuser operating systems allow multiple users to access and use the system concurrently. Each user can run multiple programs or processes simultaneously, and the operating system manages the allocation of system resources to ensure fair and efficient use of computing resources.
- **Resource Sharing:** Multiuser operating systems facilitate resource sharing among multiple users. Users can share files, directories, printers, and other resources with other users on the system, enabling collaboration and communication among users.
- **Remote Access:** Multiuser operating systems support remote access, allowing users to access the system from remote locations using network connections. Remote access protocols, such as SSH (Secure Shell) and Telnet, enable users to log in to the system and perform tasks remotely.

Gateway classes 7455 961284

Disadvantages :

- **Resource Contentions:** In multiuser environments, users may compete for system resources such as CPU time, memory, and I/O devices. Resource contention can lead to performance degradation, increased response times, and reduced system throughput.
- **Security Risks:** Multiuser operating systems are susceptible to security risks and vulnerabilities due to the presence of multiple users accessing the system concurrently.
- **User Management Overhead:** Managing user accounts, access permissions, and security policies in multiuser operating systems requires administrative overhead.
- **User Data Isolation:** Multiuser operating systems may face challenges in isolating user data and preventing unauthorized access between users.

Gateway classes 7455961284

- **Performance Degradation:** As the number of concurrent users increases, multiuser operating systems may experience performance degradation due to resource contention, increased system overhead, and scalability limitations. Users may experience slower response times and reduced system performance, particularly in shared computing environments with limited resources.
- **Complexity of Administration:** Administering multiuser operating systems can be complex due to the need for managing user accounts, access controls, system configurations, and security policies.
- **Dependency on Network Connectivity:** Multiuser operating systems may rely on network connectivity for remote access and user collaboration. Disruptions in network connectivity or network failures can prevent users from accessing the system remotely, leading to downtime, reduced productivity, and user dissatisfaction.

Gateway classes 7455 961284

6. Single Threaded and Multithreaded Operating Systems

- A program under execution is known as a process and thread is a basic unit of execution.
- A program may have a number of processes associated with it and each process can have a number of threads executing in it.
- So, thread is a basic unit of execution or a basic unit of CPU utilization.

Single-threaded Operating System : -

- In a single-threaded operating system, each process can execute only one task or instruction at a time.
- The operating system allocates CPU time to processes sequentially, allowing each process to run to completion before switching to the next process.
- Single-threaded operating systems may suffer from poor performance and inefficiency, especially in environments with multiple users or concurrent tasks.
- Example DOS

Gateway Classes 7455 961284

Multithreaded Operating System:

- In a multithreaded operating system, each process can be divided into multiple threads, each of which can execute independently.
- Threads within the same process share the same memory space and resources, allowing them to communicate and coordinate with each other more efficiently than separate processes.
- Multithreaded operating systems can exploit parallelism and concurrency, improving system responsiveness, throughput, and resource utilization.

P1

Key characteristics and advantages of multithreaded operating systems include:

- **Concurrency:** Multithreaded operating systems can execute multiple threads concurrently, allowing tasks to overlap and progress simultaneously. This concurrency enables better utilization of CPU resources and improves overall system performance.
- **Responsiveness:** Multithreading improves system responsiveness by allowing applications to remain responsive even while executing CPU-intensive tasks. Threads can handle input/output operations, user interactions, and background tasks without blocking the execution of other threads.

Gateway classes 7455 361284

- **Resource Sharing:** Threads within the same process share the same memory space and resources, facilitating efficient communication and data sharing between threads. This enables faster inter-thread communication and synchronization compared to inter-process communication.
- **Scalability:** Multithreaded operating systems are scalable, allowing applications to take advantage of multiple CPU cores and processors. Threads can be distributed across multiple CPU cores, enabling parallel execution and better performance on multi-core systems.
- **Improved Throughput:** Multithreading improves system throughput by overlapping the execution of multiple threads and reducing idle time. This results in higher overall system throughput and better utilization of system resources.
- **Examples:** Linux, Windows, Solaris, Android etc.

Gateway classes 7435 9612 84

Disadvantages:

- **Concurrency Control:** Multithreading introduces concurrency control challenges, such as race conditions, deadlock, and resource contention. Proper synchronization mechanisms, such as locks, semaphores, and monitors, are required to ensure thread safety and prevent data corruption.
- **Complexity:** Multithreaded programming can be complex and error-prone, especially when dealing with shared resources and synchronization. Developers must carefully manage thread interactions and concurrency issues to avoid bugs and ensure correct behavior.
- **Performance Overhead:** Multithreading can introduce performance overhead due to context switching, synchronization, and thread management. Excessive use of threads or improper synchronization can lead to decreased performance and scalability.

Gateway classes 7455 96 12 34

7. Multiprogramming Operating System:

- Multiple programs are allowed to reside in memory at any point of time.
- How many programs will reside in memory at a time called degree of multiprogramming.
- Multiprogramming operating systems are designed to maximize CPU utilization by allowing multiple programs to run concurrently.

Some key characteristics of multiprogramming operating systems:

- **Concurrent Execution:** Multiprogramming operating systems enable multiple programs to execute concurrently, allowing the CPU to switch between executing different processes rapidly.
- **CPU Scheduling:** PROCESS The operating system's CPU scheduler determines which process to execute next based on scheduling algorithms such as round-robin, priority scheduling, or shortest job first. This scheduling ensures fair access to CPU resources and optimizes system throughput.
- **Memory Management:** Multiprogramming operating systems manage memory allocation and deallocation dynamically to accommodate multiple processes simultaneously. Techniques such as virtual memory allow processes to access more memory than physically available by using disk storage as an extension of RAM.

Gateway classes 7455 961284

- **I/O Device Management:** The operating system manages input/output (I/O) devices efficiently to prevent them from becoming bottlenecks. I/O operations often occur asynchronously, allowing processes to continue executing while waiting for I/O completion.
- **Process Synchronization:** Multiprogramming operating systems provide mechanisms for process synchronization to coordinate the activities of concurrent processes and prevent race conditions and data corruption. Techniques such as semaphores, mutexes, and monitors ensure safe access to shared resources.
- **Interrupt Handling:** The operating system handles hardware interrupts and software exceptions to maintain system stability and responsiveness. Interrupt-driven I/O allows processes to perform other tasks while waiting for I/O operations to complete.
- **Context Switching:** Context switching is the process of saving the state of a running process, loading the state of a new process, and transferring control between them. Multiprogramming operating systems perform context switches efficiently to minimize overhead and maximize CPU utilization.

Gateway classes 7455 96 12 24

- Examples of multiprogramming operating systems include Unix/Linux, Microsoft Windows, macOS, and various real-time operating systems used in embedded systems and microcontrollers.
- Multiprogramming enables efficient utilization of CPU resources and improves system responsiveness by allowing multiple programs to execute concurrently.

Gateway classes 7455 961284

AKTU PYQs

Q.1. Define a multiprogramming system. 2016-17, 2 Marks

Q.2. Write down the different types of operating systems. 2016-17, 10 Marks

Q.3. Differentiate between (i) Interactive and Batch operating system (ii) Multiprogramming and Time sharing system 2017-18, 7 Marks

Q.4. Explain the Batch operating system with an example. 2018-19, 7 Marks

Q.5. What do you mean by multitasking? 2018-19, 2 Marks

Q.6. Discuss the difference between time sharing system and a real-time system. 2018-19, 10 Marks

Q.7. Briefly define the term real time operating system. 2021-22, 2 Marks

Q.8. Explain in detail about the multiuser and multithreaded systems. 2021-22, 10 Marks

Q.9. Explain the principal advantages of multiprogramming. 2022-23, 2 Marks

Q.10. Explain the various terms in detail: (i) Multiprocessor operating systems (ii) Real-Time system (iii) Time sharing system 2022-23, 10 Marks

Gatewayclasses 7455 9612 84

Structure of an Operating System

- The term structures of operating system typically refers to the fundamental organizational frameworks or architectures that define "how an operating system manages its resources," interacts with hardware, and facilitates user interaction.
- We know that operating system is a large and complex system and it has to be engineered carefully so that it will function properly and it can be modified easily.
- There are different structures of OS that have been followed from the beginning till now:-

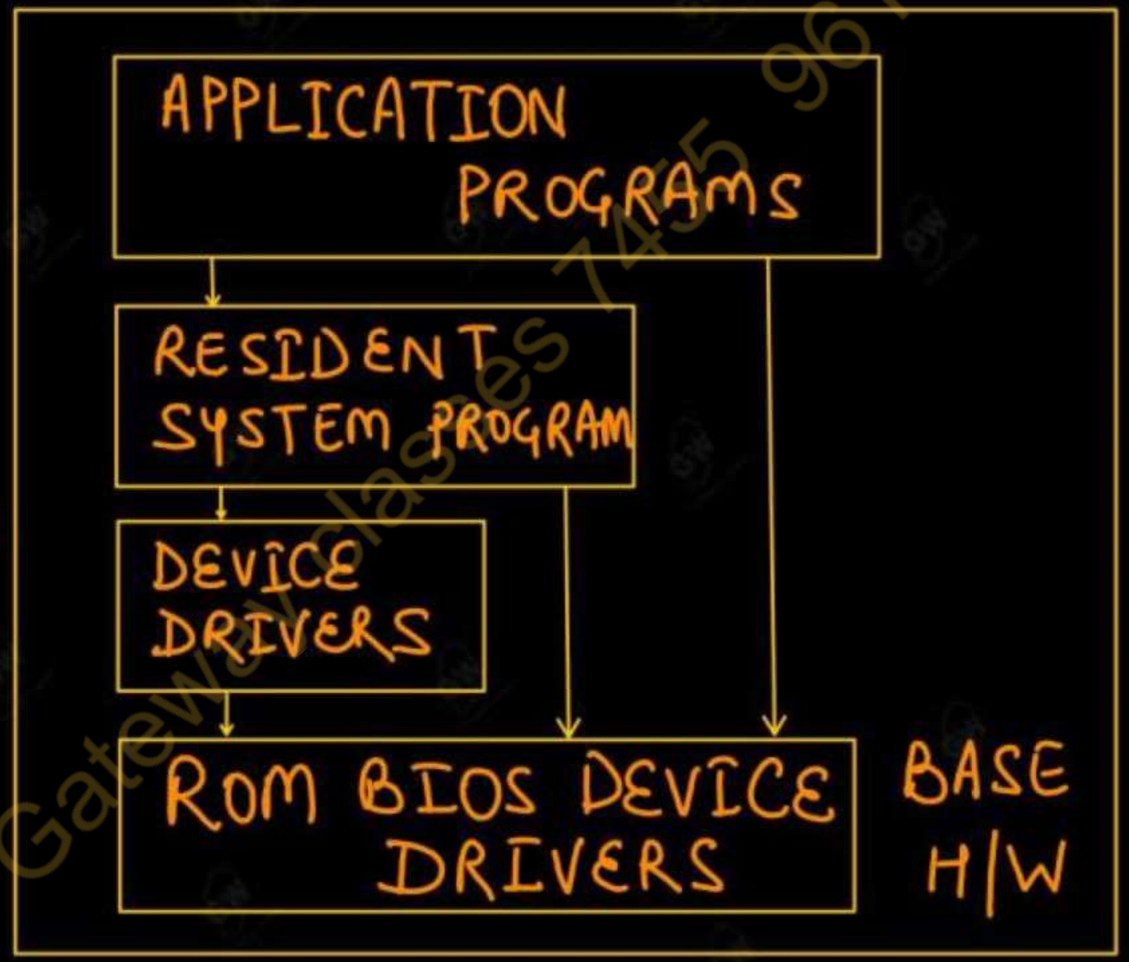
1. Simple Structure
2. Monolithic Structure
3. Layered Structure
4. Microkernel Structure
5. Modular Structure

Gateway classes 7455 961284

1. Simple Structure of the Operating System

- This structure was followed in designing the oldest operating system such as DOS.
- This structure is not very well defined.

FIG. SIMPLE STRUCTURE OF MS DOS



Gateway classes 7455 9612 84

Key characteristics of Simple Structure of Operating system:

- The bottom layer ROM BIOS is a base H/W.
- It is to be noted that the device drivers have the access of base H/W, resident system programs have the access of device drivers as well as basic H/W and application programs have the access of system programs, device drivers and basic H/W. So, base H/W can be accessed by all the things above it.
- In this structure the interfaces and levels of functionality are not well separated.
- Application programs are able to access the basic input output routines and write directly to the display and disk drivers.
- That means application programs can access the H/W directly without going the layers below it.
- This structure looks like kind of layered structure but it is not layered structure because there is another structure of operating system called layered structure.

Gateway classes 7455 9612 84

Disadvantage:-

- All layers have access to the base H/W.
- This freedom leads the MS DOS vulnerable to errant or malicious programs, causing the entire system to crash when the user program fails.
- That means if an application program is being run by the user and since it is directly accessing the base H/W, if that program fails, then that entire system is going to crash because it is having the direct access of base H/W.
- So we see that it is not protected not well structured and it is not well defined.
- So this is not a very good structure.
- We cannot blame the developers and designers for designing this because MS DOS was actually written on the Intel 8088 and this Intel 8088 does not provide dual mode or any H/W protraction.
- So the developers and the designers had no other option then to just leave this Base H/W accessible by all layers above.

Gateway classes 7455961204

2. Monolithic Structure :-

- This is also a simple structure.
- This structure was followed by earlier UNIX OS.
- In this structure we have a kernel and system programs.
- Everything is packed with a kernel only at one level that is why this structure is known as monolithic kernel.

FIG. - MONOLITHIC STRUCTURE

The Users		
Shells and commands, Compilers and Interpreters, System libraries		
System call interface to the kernel		
Signal, Terminal Handling, Character I/O System, Terminal Drivers	Files System, Swapping Block, I/O System, Disk and File Drivers	CPU Scheduling, Page Replacement, Demand Paging, Virtual Memory
Kernel Interface to the H/W		
Terminal Controllers	Device Controllers Disk & Tapes	Memory Controllers, Physical Memory

Gateway classes 7455 9612 84

Disadvantages:-

- There are two many functions packed into one level and this makes its implementation and maintenance very difficult.
- Suppose we want to modify or change at kernel level or we want to debug a certain functionality at this kernel. Suppose there is a problem with CPU scheduling. So, in order to fix that issue of CPU scheduling, we will have to touch the entire kernel so this makes it a system that is difficult to be maintained and implemented.
- If we want to add something into it, it also going to be difficult so this is the disadvange of monolithic kernel structure where too many things are packed into one level.

Gatewayclasses 7455961284

3. Layered Structure of Operating System:-

- In this structure, Operating System is divided into a number of layers.

FIG:- LAYERED STRUCTURE OF O.S.

Layered Structure of Operating System

Gateway classes 7405 9612 84

- At layer-0 there is H/W.
- The top most layer is user interface.
- In comparison to monolithic structure where everything was packed into at one level, here in this structure the functionalities have broken down into different layers.

Advantages :-

- H/W is protected form the layers above, the user interface can not directly access the H/w.
- The main advantage of this structure is that this is easy to implement & debug.
- Every layer is having different functionalities so instead of keeping these altogether it has been divided into layers.
- Let's say that one layer is having a problem. Let's say that CPU scheduling layer is having a problem. So in order to debug that we just have to look at that layer and debug it instead of going and digging into the entire OS.
- So this is one of the advantages of this layered OS structure.

Gateway classes 7455 961284

Disadvantages:-

- We have to be careful and specific in designing and deciding which will be the layer on top of a certain particular layer or which will be the layer below a particular layer because a layer can use only those layers which are below that layer i.e. only the services by its below layer can be used by a particular layer.
- For example, let's say that we are having a layer which deals with the backing storage that means the layer which ~~leaks~~ deals with disk space used by the virtual memory algorithm. So, there is a layer dealing with the backing store or Backing storage.
- This layer we have to make sure that it has to be below the layer that deals with memory management because the memory management routines needs to use the services provided by the backing storage.
- So because of this we have to make sure that the backing storage layer is below the memory management layer.
- So we need to be very careful in designing this structure.

Gateway classes 7455 961264

- An another problem of this layered structure is that this may not be very efficient as compared to the other structures.
- That is because when one layer want to use the services provided by the layers below it the request has to go down below each layer one by one and by the time the service is actually provided it may be late or it may not be very fast.
- For example, a user program from the layer N, it wants to execute an I/O operation.
- So, in order to execute this I/O operation, it has to get the services from layer 0 because I/O devices are falling under this H/W.
- So what it will do? Layer N will issue a system call in order to use the H/W.
- This system call has to go through all the layers one by one and it has to reach the layer-0.
- As the system call passes through all the layers, the parameter of the system calls may be modified and it will take time in reaching the layer 0 & then once the system call is granted then it will be able to use the I/O devices.
- So, we see that as it has to pass through all the layers one by one, it is not going to be very efficient. By the time it actually gets the I/O operations as it was requested it is going to take some time .

Gateway classes 7455 961284

4. Microkernel Structure of an Operating System:-

- We have seen that in monolithic kernel, so many functionalities were packed into the kernel, making the kernel very big and packed with number of functionalities.
- In this microkernel approach of structuring of OS, instead of having big kernel with so many functionalities, there is a small kernel.
- The approach is that we remove all non essential component form the kernel and implement them as system level or user level program.
- So, here microkernel will just provide the core functionalities of the kernel and then the other functionalities which are there such as device driver, the file server the process servers, the virtual memory all these services are implemented either as a user level or system programs.

Gateway classes 7455 9612 84

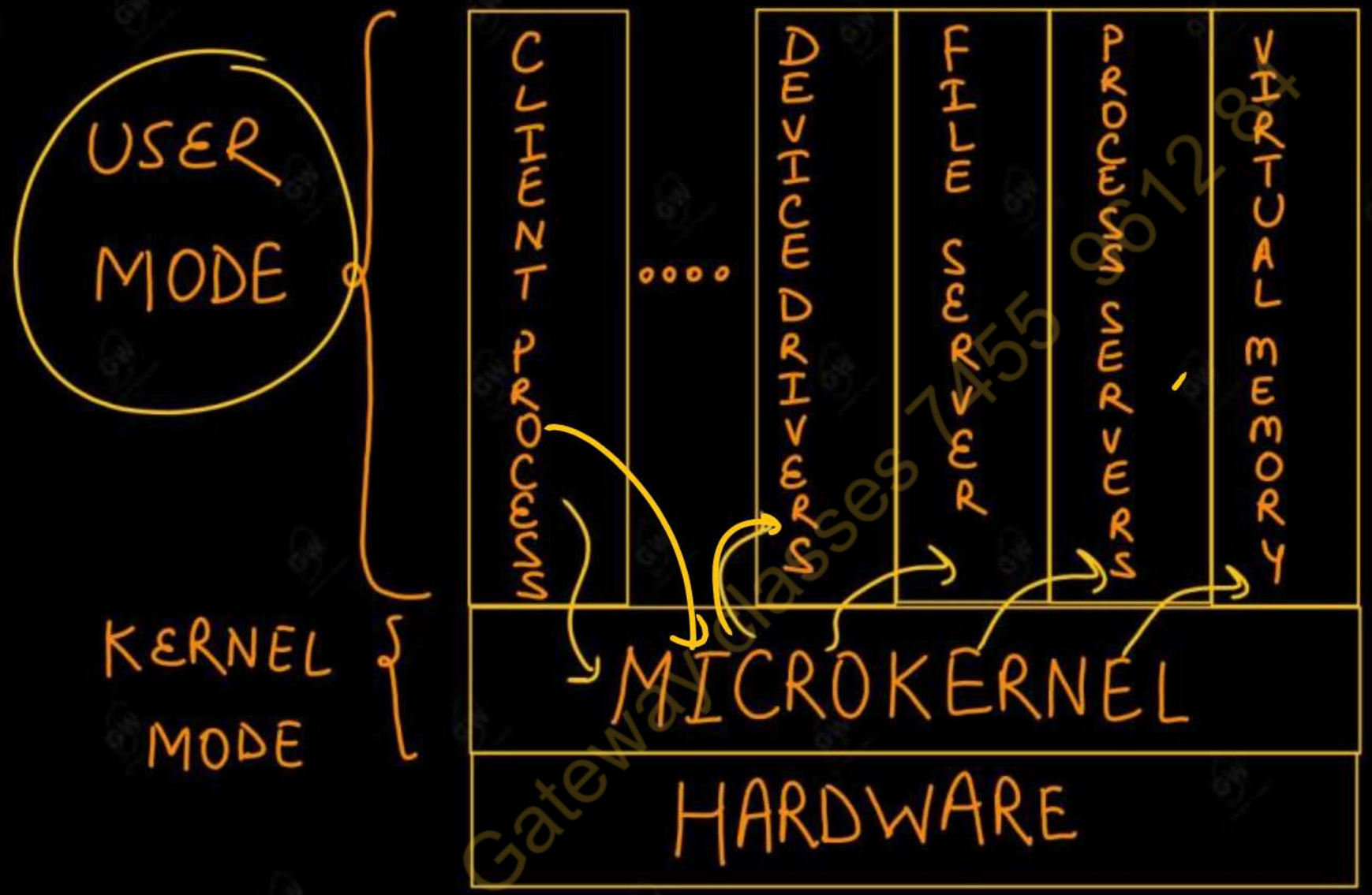


FIG.- MICROKERNEL STRUCTURE OF OPERATING SYSTEM

Gatewayclasses 7455 961284

- In this approach the main function of the microkernel is to provide a communication between the client program and these services.
- There is a client program requesting for some kind of services.
- In monolithic approach, the client program will have to ask service from the kernel and then from the kernel it has to get its services, but in this microkernel approach the client program will request something & the microkernel will just provide the communication between the client program and it will just help them to communicate with the services which are implemented as system programs.
- This communication between the client program and these system programs, which provide the services are make through something called as "Message passing".

Gateway classes 7455 9612 84

Advantages :-

- In this approach, most of the functions are executed in the user mode because the kernel is having the core functionalities.
- So, when the client program wants to access any of the services it does not actually have to be run in the kernel mode. It can be run in user mode because most of the services are there as system program.
- So if a program is executing in user mode even if the program crash. But it is running in kernel mode and the particular programs fails then the entire system is going to crash.
- So, the microkernel approach since most of the functionalities will run in the user mode, the crashing problem of the entire system is not going to happen mostly.
- So, this is the advantage of this microkernel approach of structuring the operating system.

Gateway classes 7455 96 12 84

Disadvantages:-

- Microkernel can suffer from performance decease due to the increase system function overhead.
- As the communication is done using message passing so there could be a system overhead leading to a decrease performance.

Gateway classes 7455 9612 84

5. Modular Structure of an Operating System: -

- In this structure we follow modular approach in structuring the operating system.
- This can be considered the best current methodology to structure on OS which involves using object oriented program techniques to create a modular kernel.

Gateway classes 7455 961284

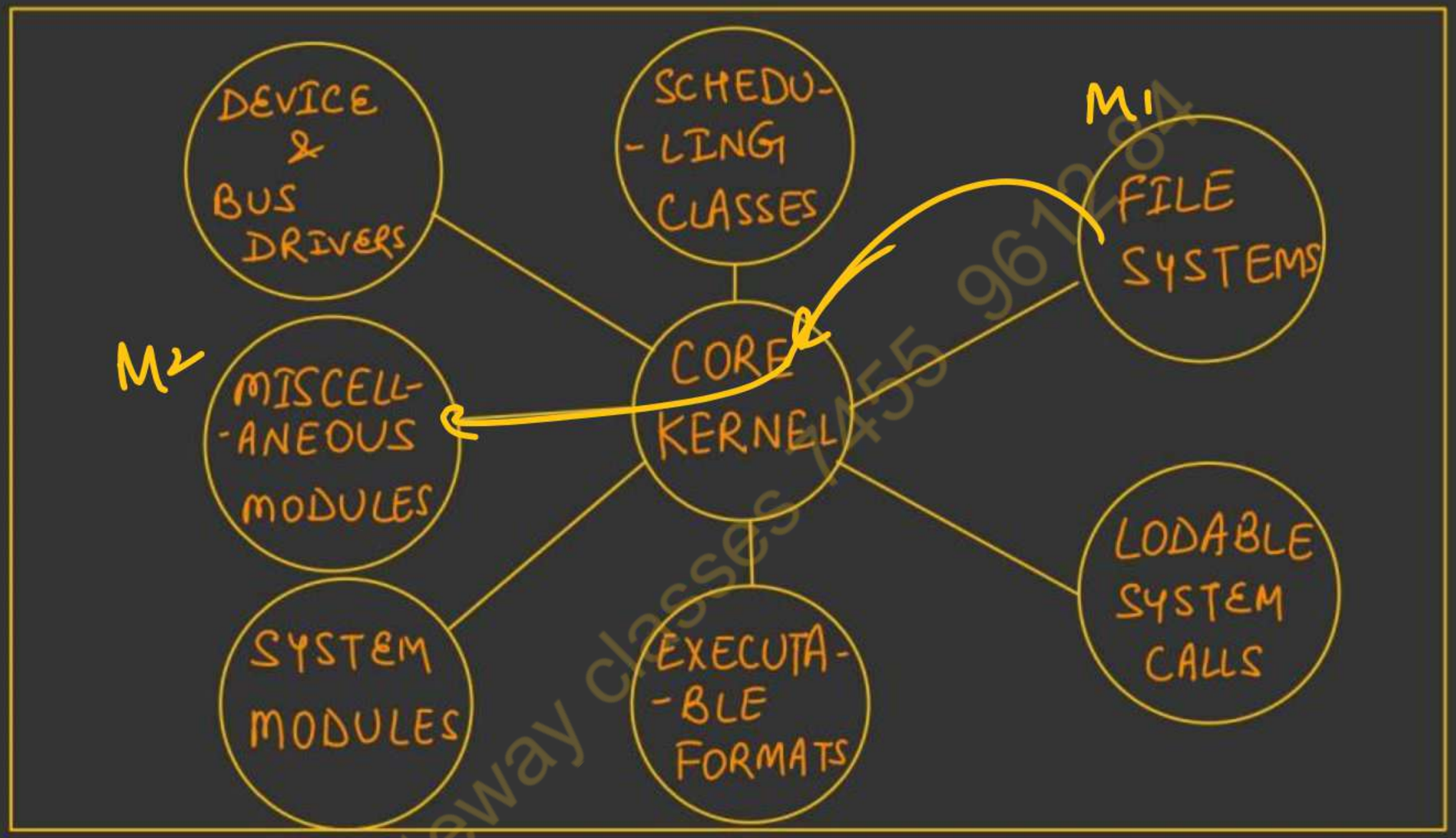


FIG :- MODULES STRUCTURE OF OPERATING SYSTEM

Gateway classes 7455 961264

Key Characteristics of Modular Structure of Operating System:

- In this approach there is a core kernel & this core kernel will have only the core functionalities of the kernel and the other functionalities are present in the forms of modules which will be loaded to the kernel either at boot time or at the run time.
- These modules may be dynamically loaded to the kernel as and when required.
- The modular approach looks something like the layered approach and also it looks a little bit like the microkernel approach.
- This structure reassembles the layered structure in such a way that each kernel section has defined protected interfaces.
- Each of the layer has a defined protected interfaces which is protected from the functionalities that we do not want them to access.
- But it is more flexible then a layered structure where any module can call any other module.

Gateway classes 7455964234

- In the layered structure we saw that when one layer want to communicate with other layer it had to go through the layers all above it .
- For example, we have seen that if layer-1 want to communicate or use some services from layer-4 then it has to go through layer-2 & layer-3 then reach layer-4.
- But in the module approach it does not have to go through those layers.
- Each of the module can communicate with the other module directly through the core kernel.
- So that is the advantages & flexibilities that are offered in modular structure as compared to the layer approach.
- So, it is more flexible and better then layered structure of OS.
- This structure looks a bit like microkernel approach is which we had a kernel which has the core kernel functionalities and other functionalities were provided as system programs above it.

Gateway classes 7455 0612 84

- In modular approach also we are having the core kernel with only the core functionalities and then other functions are loaded into it whenever necessary.
- But its advantage as compared to the microkernel approach is that in microkernel approach we need to have message passing in order to communicate between the system programs.
- But in this modular approach modules are loaded dynamically into the core kernel as & when needed.
- So, they do not need to use the message passing and the system overhead is not there so this is the advantage of modular approach as compared to layered and microkernel structure of operating system.

Gateway classes 7455 9612 84

AKTU PYQs

Q.1. Explain the layered structure approach of an operating system. Also, explain the advantages and disadvantages of the layered approach to system design. **2014-15, 5 Marks**

Q.2. Explain briefly the Layered system structure with a neat sketch. Also, explain protection and security. **2017-18, 7 Marks**

Q.3. Write about the monolithic kernel, layered, and microkernel structures of operating systems. **2018-19, 7 Marks**

Q.4. Explain in detail about the monolithic and microkernel systems. **2021-22,10 Marks**

Gateway classes 7455 961284

Difference Between Microkernel & Monolithic Kernel

Microkernel

1. **Smaller in size**

- In microkernel, the kernel space only contains the most important part i.e. the interposes communication and the scheduling part while all other components are present in the user's space like memory management, device drive, network stack etc.

2. **Slower in execution**

- In the microkernel the communication between components is done by using message passing which is pretty much slower as compared to the procedure calls.

Monolithic Kernel

1. **Larger in size**

- In monolithic kernel all the components are present in the same address space within the kernel. Components like network stack, memory management, device drivers, interposes communication, CPU scheduling etc. all components are present in only one space.

2. **Faster in execution**

- Because the monolithic kernel has all the components in same address so no message passing is required for communication i.e. it is faster to execute any operation.

Gateway classes 7455 961284

3. Less number of line of code in kernel

- In the microkernel the number of lines are less because of smaller size of the kernel.

4. Easy debugging & management

- Because the kernel is smaller the number of lines of code will be smaller so the debugging and management of a smaller code is easier than the larger one.

5. Easy to add new functionality

- Because while adding a new functionality, suppose we want to add a new functionality f1 to the microkernel then we need to update the codes of kernel and if we have

3. Larger number of line of code in kernel

- Monolithic kernel has kernel space and so the number of lines will also be huge.

4. Debugging is difficult

- Because the kernel is big the number of lines of code will be larger so the debugging and management of a larger code is difficult than the smaller one.

5. It is difficult to add new functionality

Gateway classes 7455 961284

lesser number of lines in the kernel then number of lines we need to manage on update in order to add this functionality will be lower and so this will be very much less hectic then the one in the monolithic kernel which has the larger number of lines. Because a large number of lines have to be managed in order to add functionality. So it is easy with microkernel.

6. If one component crash others keep running

- Because all the components are not on same space so cascading error effect is not encountered.

6. If one component crash entire system crashes

- Because all the components are present in same address space for example consider the following 'C' program-

Gateway classes 7455 9612 84

f1(C)	Function-1
f2(C)	Function-2
f3(C)	Function-3
main( )	Function-4

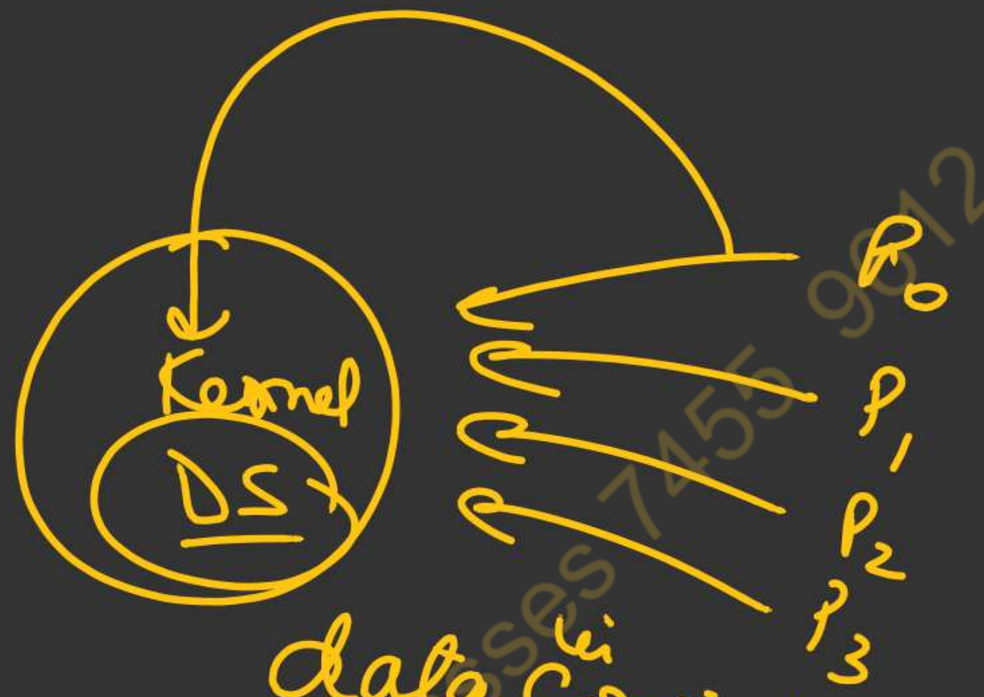
- Suppose there is any error with any function say in f2(C) then the entire program show the compilation error.
- If we write these functions in different programs then the correct program with correct function will run and incorrect will fail.

Gateway classes 7455 961284

Reentrant Kernel

- A reentrant kernel, as its name implies, permits multiple processes to execute concurrently in kernel mode without encountering inconsistencies among the kernel's data structures.
- That is many process or threads can be executed at the same time but without altering one another.
- For example processes P1 and P2 are being executed in kernel mode in parallel then the execution of process P1 will not affect the execution of process P2 and vice versa.
- It is not a type of kernel, it is a feature of Operating System, it is not part of structure, it is that kernel is type of reentrant or not.
- Suppose there is a kernel space and process P1, P2, P3 & P4 wish to access the data structure available with kernel.

Gatewayclasses 7455 964284



data consistency

A = 5 6

Read ( )

B

Gateway classes 7455 9612 84

- So, if all process will be allowed to access the Data Structure of kernel then it may cause the problem of data inconsistency. But reentrant kernel is that feature of Operating system which allows execution of multiple process in kernel mode but avoid the problem of data inconsistency by using some special characteristics.
- For example, process P1 is being executing on processor for some time & then need to access/read some data from disk/memory.
- During this read( ) operation process P1 will release the processor for a particular time.
- In this way during that duration another process P2 will get the chance of execution through processor in kernel mode.
- That is in a reentrant kernel a process waiting on a read( ) call may decide to release CPU to a process which is waiting for execution in kernel mode.

Gateway classes 7455 961284

Why a kernel is made Reentrant?

- Lets suppose a kernel not of reentrant nature.
- Lets assume that a process is executing in kernel mode and accessing a kernel data structure and some global values associated with it.
- Suppose the process name is 'A'.
- Suppose there is a global variable X and process A want to read this variable X.
- But just before process A tries to use the value of global variable X, a context switch to process 'B' happens.
- Now this process 'B' tries to access the value of this global variable X and suppose decrements it.
- Another context switch happens and process 'A' comes back into execution.
- Since process 'A' does not know that process 'B' has already decremented the value of global variable X, it tries to use this value again.
- So here is the catch, process 'A' sees two different values of global variable X as the value was changed by another process 'B' So, this is the problem of data inconsistency so, if more then one processes is being allowed in a kernel spaces then the kernel must be reentrant in nature.

Process – A → Read X = 5

B ⇒ 4

Gateway classes 7455 961264

How to make a kernel reentrant?

- Write kernel functions that modify only the local variables and do not alter the global variable or data structures. Such type of functions are known as "Reentrant Functions". Write the code which can manipulate only local data structure i.e. can not modify to the global data structure.
- Relying solely on the use of reentrant functions within a kernel isn't always practical.
- Another approach involves employing a "locking mechanism" to ensure that only one process can access a non-reentrant function at any given time.
- Since implementing reentrant functions is more related to a good programming and the locking mechanisms are related with the concept of synchronization.

Gateway classes 7455 9612 84

Services Provided by an Operating System

The Operating System provides certain services to the users. Some major services are as follows:

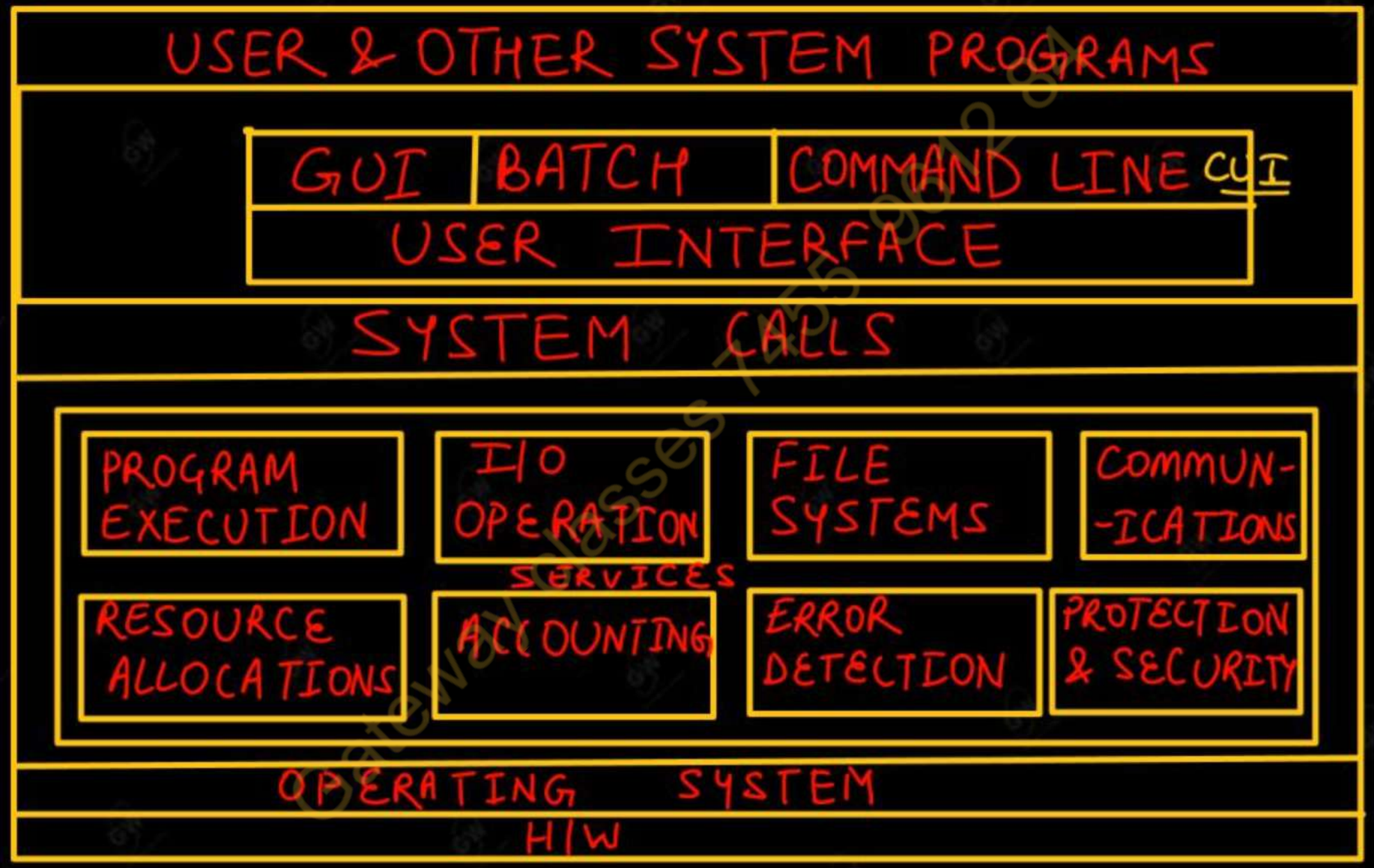
- **Interface:** 1
  - ✓ The Operating System provides the interface facility to the user.
  - ✓ Interfaces such as Graphical User Interface (GUI), Command prompt interface, Touch interface
- **Program Execution:** 2
  - ✓ The Operating System is responsible for the execution of all types of programs whether it is user programs or system programs.
  - ✓ The Operating System utilizes various resources available for the efficient running of all types of functionalities.
- **Input / Output Operations:** 3
  - ✓ The Operating System is responsible for handling all sorts of inputs, i.e., from the keyboard, mouse, desktop, etc.
  - ✓ The Operating System does all interfacing most appropriately regarding all kinds of Inputs and Outputs.

Gateway classes 7455 9612 84

- **File System Manipulation:** 4
- The Operating System is responsible for making decisions regarding the storage of all types of data or files, i.e., floppy disk/hard disk/pen drive, etc.
- The Operating System decides how the data should be manipulated and stored.
- **Communication (Inter-process Communication):** 5
  - ✓ It provides communication facility among the processes to execute any operation.
  - ✓ Let suppose printing of a word document via printer
- **Error Detection and Handling:** 6
  - ✓ The Operating System is responsible for the detection of any type of error or bugs that can occur while any task or during the execution of any instruction.
  - ✓ The well-secured OS sometimes also acts as a countermeasure for preventing any sort of breach of the Computer System from any external source and probably handling them.

Gateway classes 7455 961284

FIG:- A VIEW OF OS SERVICES



Gateway Classes 7455 9002 81

Resource Allocation: 7

- ✓ The Operating System ensures the proper use of all the resources available by deciding which resource to be used by whom for how much time.
- ✓ All the decisions are taken by the Operating System.

Accounting: 8

- ✓ The Operating System tracks an account of all the functionalities taking place in the computer system at a time.
- ✓ All the details such as the types of errors that occurred are recorded by the Operating System.
- ✓ Services like creating, opening, and closing accounting files, such as spreadsheets, databases, or documents, Organizing accounting data into directories or folders for easy access and management etc.

Gateway Classes 7455 9612 84

Information and Resource Protection: 9

- ✓ The Operating System is responsible for using all the information and resources available on the machine in the most protected way.
- ✓ The Operating System must foil an attempt from any external resource to hamper any sort of data or information.

System Services: 10

- ✓ The operating system provides various system services, such as printing, time and date management, and event logging.
- All these services are ensured by the Operating System for the convenience of the users to make the programming task easier.

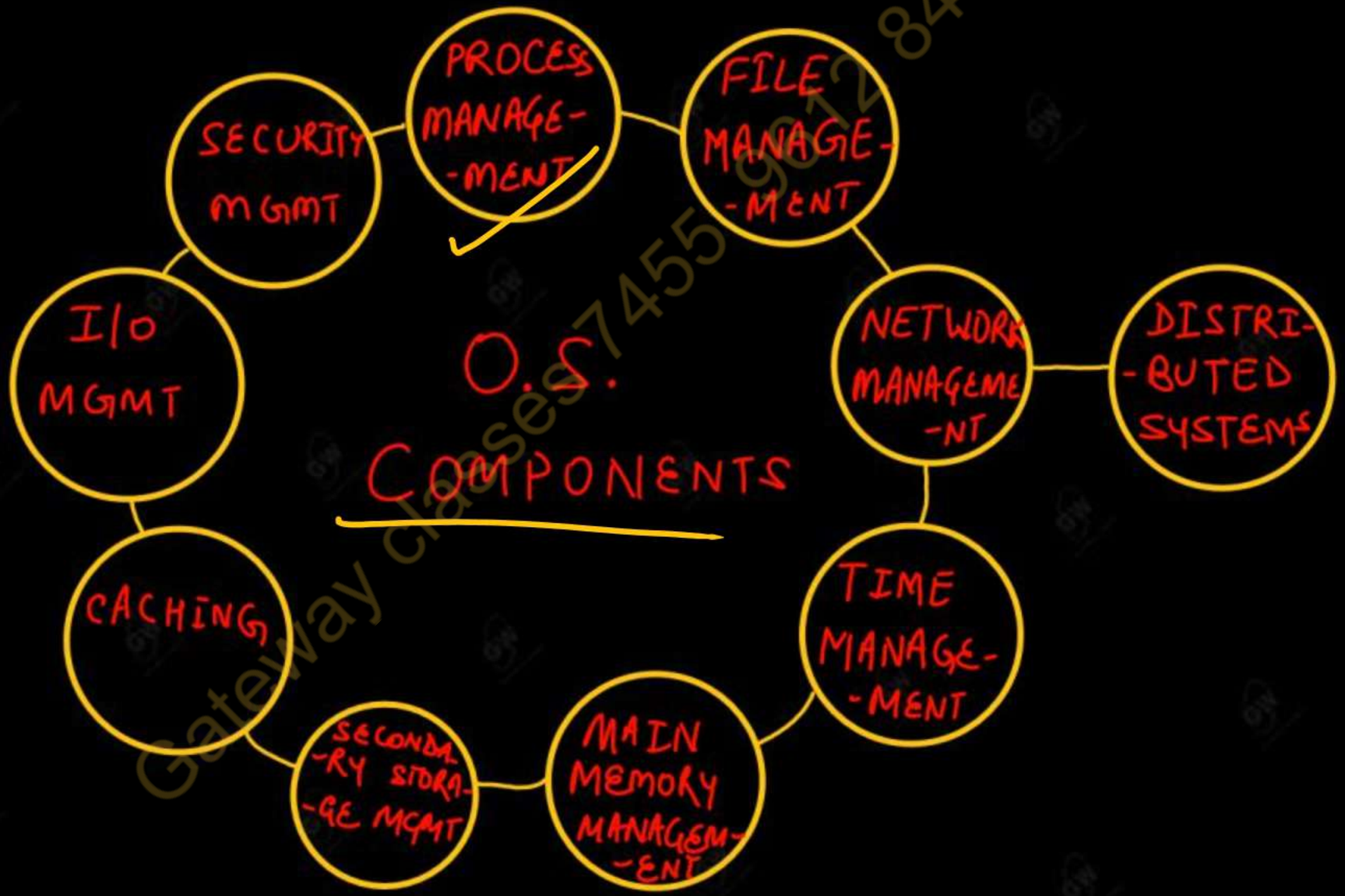
Gateway classes 7455 9612 84

Components of an Operating System

- An Operating system is an interface between users and the hardware of a computer system.
- It is a system software that is viewed as an organized collection of software consisting of procedures and functions, providing an environment for the execution of programs.
- The operating system manages resources of system software and computer hardware resources.
- It allows computing resources to be used in an efficient way.
- Programs interact with computer hardware with the help of operating system.
- A user can interact with the operating system by making system calls or using Operating System commands.

Gateway classes 7455 961284

Some Major Components of an Operating System



Gateway classes 7455961284

1. Process Management:

- A process is a program in execution.
- It consists of the followings:
  - ✓ Executable program
  - ✓ Program's data
  - ✓ Stack and stack pointer
  - ✓ Program counter and other CPU registers operational registers
  - ✓ Details of opened files
- A process can be suspended temporarily and the execution of another process can be taken up.
- A suspended process can be restarted later.
- Before suspending a process, its details are saved in a table called the process table so that it can be executed later on.
- An operating system supports two system calls to manage processes - Create and Kill-

Gateway classes 7455 9612 84

- ✓ **Create** is a system call used to create a new process.
- ✓ **Kill** system call used to delete an existing process.
- ❑ A process can create a number of child processes.
- ❑ Processes can communicate among themselves either using shared memory or by message-passing techniques.
- ❑ Two processes running on two different computers can communicate by sending messages over a network.

Gateway classes 7455 9612 84

2. Files Management:

- Files are used for long-term storage.
- Files are used for both input and output.
- Every Operating System provides a file management service.
- This file management service can also be treated as an abstraction as it hides the information about the disks from the user.
- The operating system also provides a system call for file management.
- The system call for file management includes -

✓ File creation

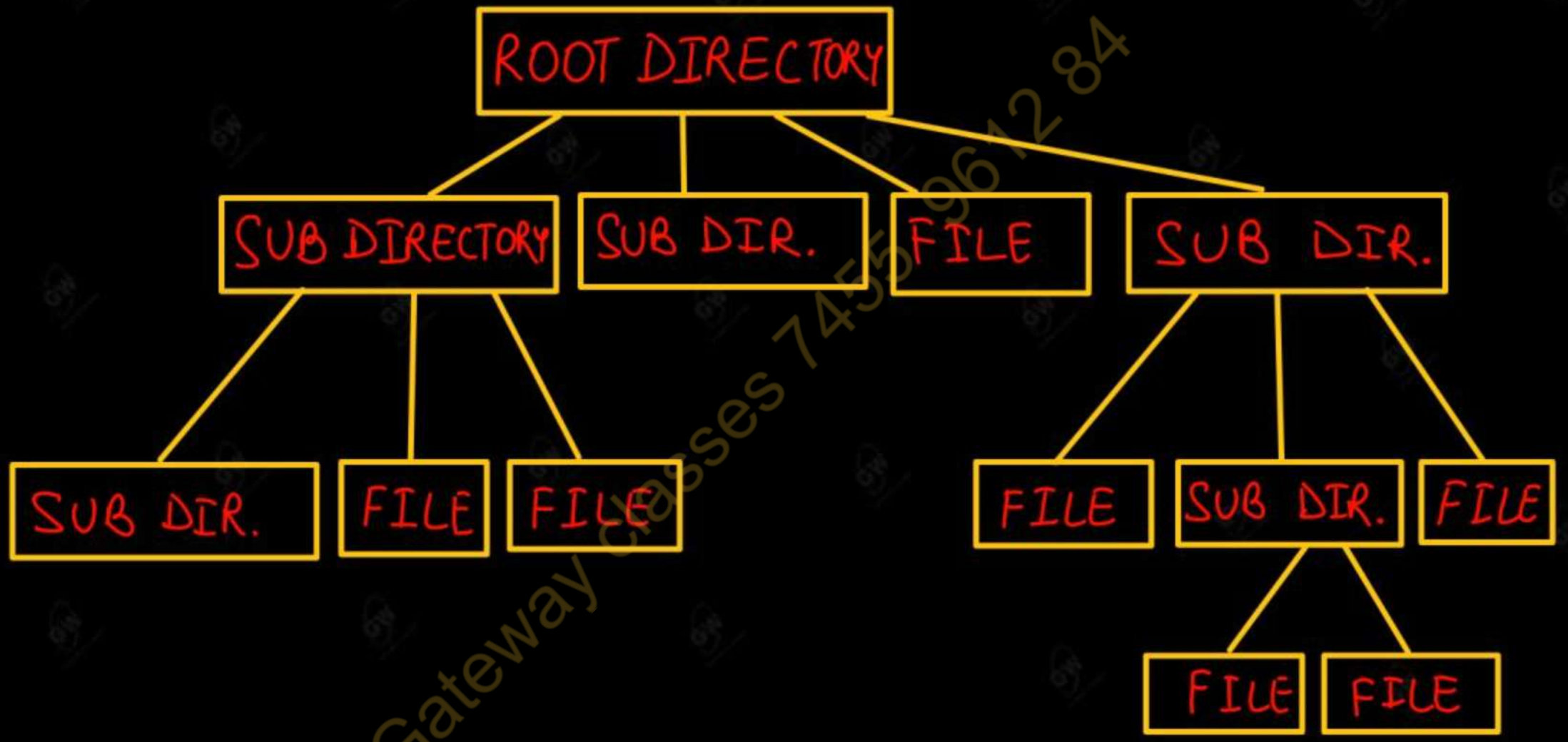
✓ File deletion

✓ Read and Write operations

- Files are stored in a directory.
- System calls provide to put a file in a directory or to remove a file from a directory.
- Files in the system are protected to maintain the privacy of the user.

Gateway classes 7455 3612 84

- Below shows the **Hierarchical File Structure directory.**



Gateway classes 7455961284

3. Command Interpreter:

- There are several ways for users to interface with the Operating System.
- One of the approaches to user interaction with the Operating System is through commands.
- Command interpreter provides a command-line interface. DOS
- It allows the user to enter a command on the command line prompt (cmd).
- The command interpreter accepts and executes the commands entered by a user.
- For example, a shell is a command interpreter under UNIX.

Gateway classes 7455 5612 84

4. System Calls:

- System Calls provide an interface to the services made by an Operating System.
- The user interacts with the Operating System programs through System Calls.
- These calls are normally made available as library functions in high-level languages such as C, Java, Python etc.
- It provides a level of abstraction as the user is not aware of the implementation or execution of the call made.
- Different hardware and software services can be availed through System Calls.
- System Calls are available for the following operations:

✓ Process Management

✓ Memory Management

✓ File Operations

✓ Input / Output Operations

Gateway Classes 7455 961284

5. Signals:

- Signals are used in the Operating Systems to notify a process that a particular event has occurred.
- Signals are the software or hardware interrupts that suspend the current execution of the task.
- Signals are also used for inter-process communication.
- A signal follows the following pattern:
  - ✓ A signal is generated by the occurrence of a particular event, it can be the click of the mouse, the execution of the program successfully or an error notifying, etc.
  - ✓ A generated signal is delivered to a process for further execution.
  - ✓ Once delivered, the signal must be handled.
- The signal causes temporarily suspends the current task it was processing, saves its registers on the stack, and starts running a special signal handling procedure, where the signal is assigned to it.

Gateway classes 7455 961284

6. Network Management:

- In today's digital world, the complexity of networks and services has created modern challenges for IT professionals and users.
- Network management is a set of processes and procedures that help organizations to optimize their computer networks.
- Mainly, it ensures that users have the best possible experience while using network applications and services.
- Network management is a fundamental concept of computer networks.
- Network Management Systems is a software application that provides network administrators with information on components in their networks.
- It ensures the quality of service and availability of network resources.
- It also examines the operations of a network, reconstructs its network configuration, modifies it for improving performance of tasks.

Gateway classes 7455 9612 84

7. Security Management:

- The security mechanisms in an Operating System ensure that authorized programs have access to resources, and unauthorized programs have no access to restricted resources.
- Security management refers to the various processes where the user changes the file, memory, CPU, and other hardware resources that should have authorization from the Operating System.

Gateway classes 7455961284

8. I/O Device Management:

- The I/O device management component is an I/O manager that hides the details of hardware devices and manages the main memory for devices using cache memory, buffering and spooling.
- This component provides a buffer cache and general device driver code that allows the system to manage the main memory and the hardware devices connected to it.
- It also provides and manages custom drivers for particular hardware devices.
- An I/O device management component allows highly efficient resource utilization while minimizing errors and making programming easy on the entire range of devices available in their systems.

Gateway classes 7455 961284

9. Secondary Storage Management:

- Broadly, the secondary storage area is any space, where data is stored permanently and the user can retrieve it easily.
- Other spaces, such as CD-ROM/DVD drives, flash memory cards, and networked devices, also provide secondary storage for data on the computer.
- The computer's main memory (RAM) is a volatile storage device in which all programs reside, it provides only temporary storage space for performing tasks.
- Secondary storage refers to the media devices other than RAM (e.g. CDs, DVDs, or hard disks) that provide additional space for permanent storing of data and software programs which is also called non-volatile storage.

Gateway classes 7455 9612 84

10. Main Memory Management:

- Main memory is a flexible and volatile type of storage device.
- It is a large sequence of bytes and addresses used to store volatile data.
- Primary memory is costly and low in terms of storage as compared to secondary storage devices.
- Whenever computer programs are executed, it is temporarily stored in the main memory for execution.
- Later, the user can permanently store the data or program in the secondary storage device.

Gateway Classes 7453 9612 84

AKTU PYQs

Q.1. Discuss various operating system components.

2014-15, 5 Marks, 2017-18, 7 Marks

Q.2. Explain the services provided by the operating system.

2016-17, 10 Marks, 2018-19, 10 Marks, 2021-22, 10 Marks

Gateway classes 7455 9612 84

Download Gateway Classes Application
From Google Play store
Link in Description

Gateway classes 7453961284

Thank You