Unit 1: Operating System Concepts

1.1 Evolution of Operating System & History
1.2 Need of an Operating System
1.3 Single User & Multi-User Operating System
1.3.1 Types of OS and their advantages & disadvantages
1.3.2 Batch OS, Distributed OS, Multi-tasking OS
1.3.3 Realtime OS, Mobile OS
1.4 Elements of an Operating System
1.5 Operating System as a Resource Manager

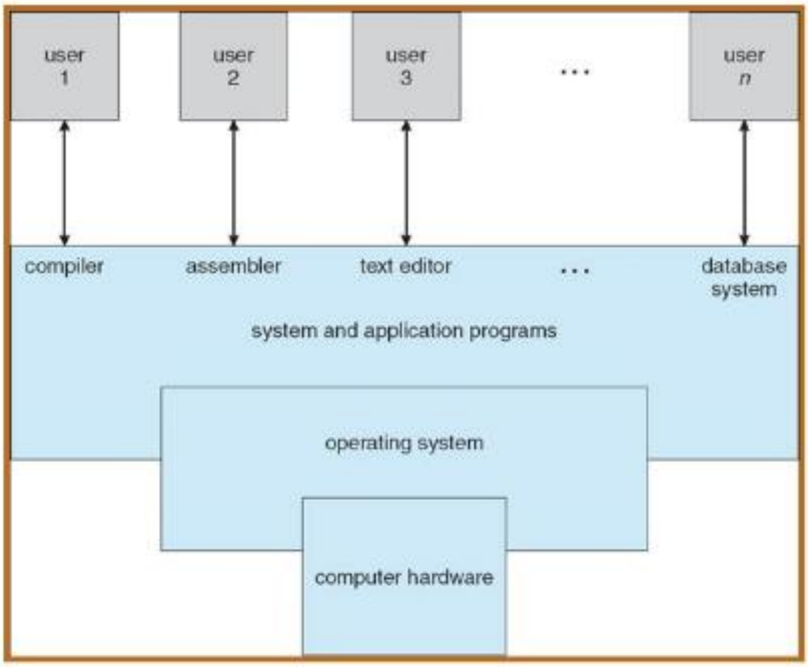
1.1 Evolution of Operating System & History

- An operating system is a program that acts as an intermediary between a user of a computer and the computer hardware.
- Purpose of operating system
  - To provide an environment where users can execute programs and applications.
  - To allow users to access the computer resources.
  - To provide a simple interface to the user so he can use the computer system efficiently.
- Goal of operating system
  - To make the computer system convenient for the user.
  - To use the computer hardware efficiently.

How operating system is important part of the computer system?

OR

Explain the components of the computer system.



The computer system can be divided into four main components:

1) User

2) Application program

3) Operating system

4) Hardware

1) **Hardware**: The hardware comes at the lowest level. It contains various kinds of physical devices. Like processor, memory, keyboard, mouse, monitor etc…

2) **Operating system**: The next level is for the operating system. It manages all the underlying hardware. It hides complex details of hardware from the user. Thus, it provides a simple interface between application program and hardware. It also works as a resource allocator. It controls input-output devices and users.

3) **Application Programs**: These are the applications through which users can communicate with the computer's internal system. They use different functionalities an operating system provides to perform their tasks. Application programs include airline reservation systems, web-browser, games, MS Office, etc…

4**) Users:** Users are at the top level. Users interact with the system by using application programs to perform particular tasks.

History of operating systems

From the beginning, computers, and operating systems have evolved a lot. Very early, Charles Babbage, an English mathematician, designed a true digital computer. He spent most of his life making the machine work properly but remained unsuccessful. As an important outcome of these efforts, he realized the need for software. After that unsuccessful attempt at Babbage's history was as follows:

1) **The First Generation (1945 - 1955):**

- Hardware: Vacuum tubes and plug boards.
- Neumann and other people help in building the calculating engine.
- No operating system, No Programming languages.
- Introduction of punch cards.

2) **The Second Generation (1955 – 1965):**

- Hardware: Transistors.
- Clear separation between designers, builders, operators, programmers and maintenance personnel.
- Machines were called Mainframes.
- Batch operating system took birth.

3) **The Third Generation (1965 – 1980):**

- Hardware: Integrated Circuits (ICs).

- Multiprogramming operating systems and variations such as time-sharing, Interactive, and Multitasking operating systems came into the picture.

4) **The Fourth Generation (1980 – Present):**

- Hardware: Large Scale Integration (LSI) circuits.
- Personal Computers evolved.

- In early 1980, IBM designed the IBM PC. They wanted software so that their system could work.
- So, people from IBM contacted Bill Gates to get some help. Gates suggested them to meet Kildall.
- At that time Kildall was running a company named Digital Research.
- Digital Research was the world's leading operating system company at that time.
- But, unfortunately, kildall refused to meet IBM, and the deal between them failed. This decision of kildall is known as the worst business decision.
- After that, IBM again contacted Gates. Now, Gates bought DOS (Disk Operating System) from a company named Settle Computer Products. Then, Gates hired a person, designer of DOS, Tim Peterson, to modify the DOS. This revised DOS is known as MS-DOS.
- IBM used this MS-DOS and became popular in the market of PCs. And history knows very well about what happened with the minded man, Mr. Bill Gates…!!!
- Later on Apple came up with a system that provided user friendly GUI. And OS like Windows 95, 98, and XP are invented.
- In parallel to this UNIX and Linux also got a good evolution.

Features of an Operating system

The operating system has the following features:

1. **Convenience:** An OS makes a computer more convenient to use.
2. **Efficiency:** An OS allows the computer system resources to be used efficiently.
3. **Ability to Evolve:** An OS should be constructed in such a way as to permit the effective development, testing, and introduction of new system functions at the same time without interfering with service.
4. **Throughput:** An OS should be constructed so that It can give maximum **throughput** (Number of tasks per unit time).

Functionalities of the Operating System:

- **Resource Management:** When parallel accessing happens in the OS when multiple users are accessing the system the OS works as a Resource Manager, its responsibility is to provide hardware to the user. It decreases the load in the system.
- **Process Management:** It includes various tasks like **scheduling and termination** of the process. It is done with the help of **CPU Scheduling** algorithms.
- **Storage Management:** The **file system** mechanism used for the management of the storage. **NIFS**, **CFS**, **CIFS**, **NFS**, etc. are some file systems. (Common interface file

system, common internet file system, and networked file system) All the data is stored in various tracks of Hard disks that are all managed by the storage manager. It included a **Hard Disk**.

- **Memory Management:** Refers to the management of primary memory. The operating system has to keep track of how much memory has been used and by whom. It has to decide which process needs memory space and how much. OS also has to allocate and deallocate the memory space.
- **Security/Privacy Management:** Privacy is also provided by the Operating system using passwords so that unauthorized applications can't access programs or data. For example, Windows uses **Kerberos** authentication to prevent unauthorized access to data.

1.2 Need of an Operating System

Why Operating system is required? Or What is the goal of the operating system? Or What is the objective of the Operating system? Or Which are the services provided by the operating system?

User point of view

1) Program execution
The main purpose of OS is to provide an efficient and convenient environment for the execution of programs. So, an OS must provide various functions for loading a Program into the main memory; executing it; and after execution, terminating it.

2) Program Development
The OS provides a variety of facilities and services such as editors and debuggers to assist the programmer in creating programs. Typically, these services are in the form of utility programs that, while not strictly part of the core OS are referred to as application program development tools.

3) Controlled access to files
For file access, the OS must reflect a detailed understanding of not only the nature of the I/O device but also the structure of the data contained in the files on the storage medium. Further, in the case of a system with multiple users, the OS may provide a protection mechanism to control access to the files.

4) Access to I/O devices
A running program needs I/O operations for reading input data, and for outputting the results. This I/O maybe with a file or with a device. As users cannot control I/O devices directly for security reasons, OS provides services for I/O operations.

5) Error detection
An error may be in user programs, CPU and memory hardware, or in I/O devices. Operating System detects such errors and makes from them. It also provides some error recovery mechanisms.

System Point of View

6) Resource Allocation

When multiple users are sharing the same machine, or when multiple jobs are running simultaneously, there is a need for fair allocation of resources among them. The operating system does this.

7) Accounting

Accounting is the process of keeping information about which user uses which resources, and for what duration of time. Such information can be used to bill the users in a multi-user environment or to get usage statistics to make plans.

8) Protection

The operating system ensures that all access to system resources is controlled. Also, security from outsiders is important.

1.3 Single User & Multi User OS

Single-user operating system:

- A single-task system is developed for use with a computer or electronic device that will only run one application at a time.
- This type of OS is typically used on devices such as wireless phones and two-way messaging devices.
- A single-task, single-user operating system can only run one program or application at a time, and so is not as useful for a computer or other device intended to run multiple programs at once.
- The Palm OS for Palm handheld computers is a good example of a modern single-user, single-task operating system.

Multi-user operating system:

- It allows multiple users to use the resources on a single computer simultaneously or at different times.
- It allows many different users to take advantage of the computer's resources simultaneously.
- Some operating systems permit hundreds or even thousands of concurrent users.
- Examples of multi-user operating systems are Linux, UNIX, Windows 2000, VMS, and mainframe operating systems etc…

Self-study

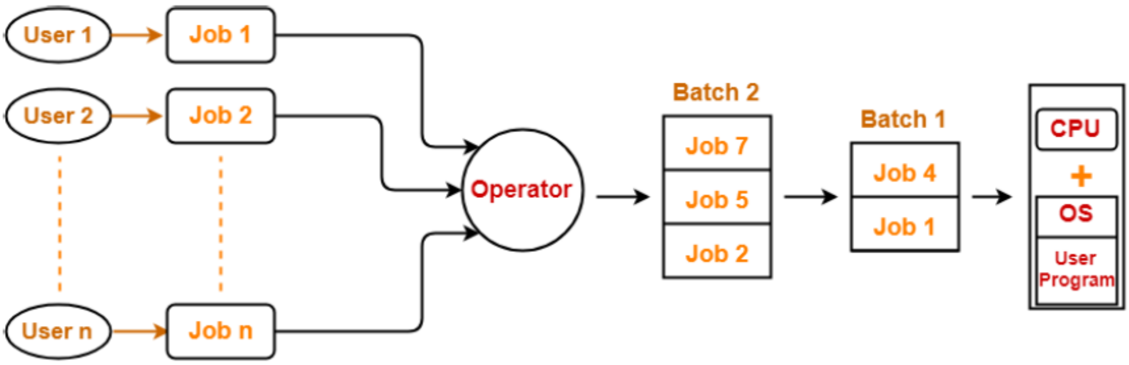
Difference between Single-user & Multi-user OS

1.3.1 Types of Operating Systems

1. **Batch Operating System**

- Early computers operated from consoles, using input devices like card readers and tape drives, and output devices like line printers and punched cards, with users preparing jobs.

- Then, he submits the job to the computer operator.
- Operator collects the jobs from different users and sorts the jobs into batches with similar needs.
- Then, the operator submits the batches to the processor one by one.
- All the jobs of one batch are executed together.
- The operating system, resident in memory, automatically transfers control between jobs, allowing operators to batch similar jobs for faster processing. CPU idles due to slower I/O devices, but disks sometimes improve performance.



Batch Operating System

Advantages

- It saves the time that was being wasted earlier for each process in context switching from one environment to another environment.
- No manual intervention is needed.

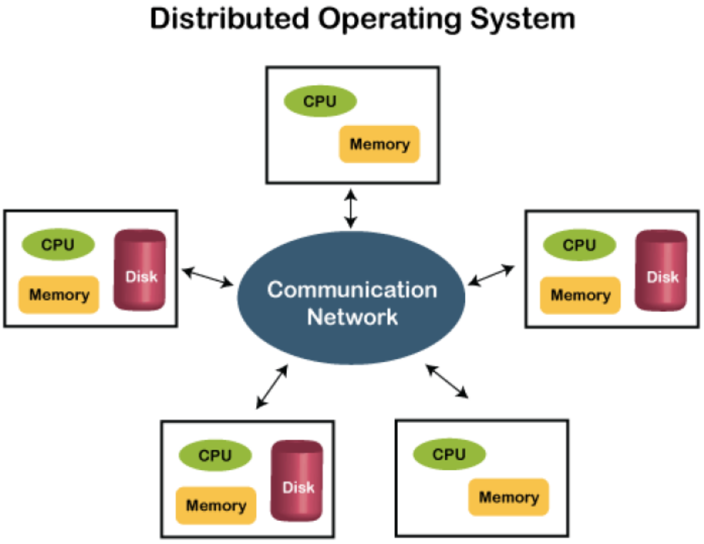
Disadvantages

- Priority cannot be set for the jobs.
- Batch operating system may lead to starvation.
- CPU may remain idle for a long time.
- There is a lack of interaction between a user and his job.

2. Distributed Operating System

- Distributed system (Loosely coupled systems) consists of a collection of independent Computers, connected through a network, which enables computers to coordinate their activities and to share the resources of the system so that users recognize the system as a single, integrated computing facility.
- They do not share memory and clocks. Each processor has its local memory.
- The processors communicate with one another through various communication lines, such as high-speed buses or telephone lines.
- Access to a shared resource increases computation speed, functionality, data availability, and reliability.
- Network change by the protocols used the distances between nodes and the transport media.

- These networks also vary in their performance and reliability. May be either client-server or peer-to-peer systems.



Key Features of Distributed Operating Systems

i. **Transparency**: DOS hides the complexity of distributed systems, making multiple systems appear as a single unit to users.

ii. **Fault Tolerance**: Distributed systems can be resilient; if one node fails, others can take over its tasks.

iii. **Resource Sharing**: Distributed OS allows for the sharing of resources, such as files, data, and devices, across the network.

iv. **Scalability**: It supports the addition of new nodes easily, which helps with scaling the system.

v. **Concurrency:** Multiple nodes can work on tasks simultaneously, improving efficiency and performance.

Advantages

i. **Reliability**: If one node fails, others can take over, enhancing the system's resilience.

ii. **Resource Sharing**: DOS allows for shared access to resources, like files and printers, across the network, leading to more efficient resource utilization.

iii. **Scalability**: New computers or devices can be easily added to the system without major disruption.

iv. **Performance**: Distributed processing can speed up computations by dividing tasks among multiple machines.

v. **Flexibility**: DOS allows integration of different types of hardware and operating systems, making it more adaptable to changes and needs.

Disadvantages

i. **Complexity**: DOS is more complex to design, implement, and maintain, requiring specialized software and expertise.

ii. **Security Risks**: With multiple systems and shared resources, the risk of unauthorized access, data breaches, and cyberattacks increases.

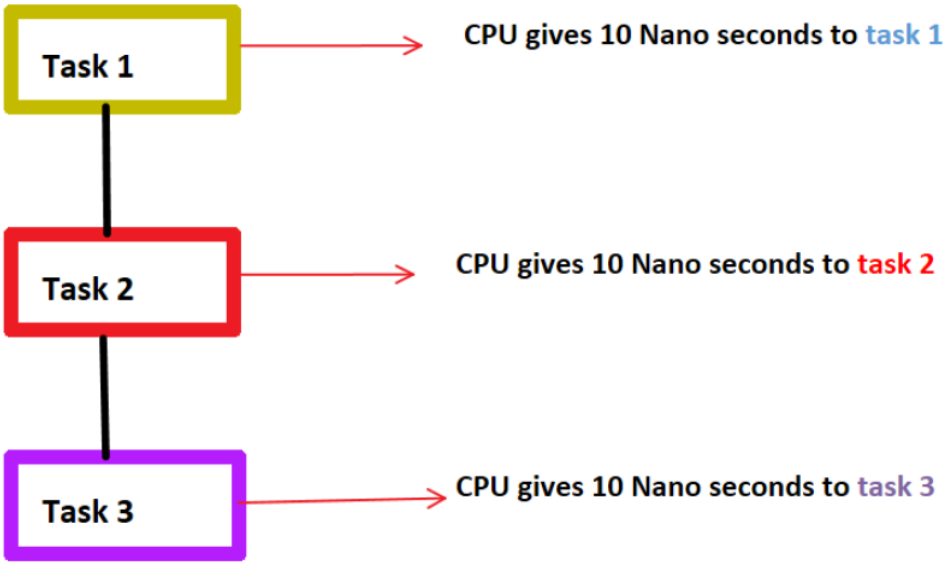
iii. **Network Dependency**: DOS performance depends heavily on the underlying network; network latency or failures can severely impact operations.

iv. **Synchronization Issues**: Coordinating tasks and data across multiple nodes requires complex synchronization mechanisms, which can lead to latency.

v. **Cost**: Setting up a distributed system can be expensive due to the need for high-quality hardware, software, and network infrastructure.

3. Multi-tasking Operating System

- Time sharing, or multitasking, is an extension of multiprogramming, allowing users to interact with each program while it is running.
- Multi-programmed systems minimize CPU idle time but struggle with proper user interaction.
- In time-sharing systems, the CPU executes multiple jobs by switching among them frequently, allowing users to directly instruct the OS or program.
- The response time should be short, typically less than one second.
- This creates the impression that the entire computer system is dedicated to one user, even though it is shared among many users.
- Time-shared OS uses CPU scheduling and multiprogramming to provide each user with a small portion of the computer.
- This requires several jobs to be kept simultaneously in memory, leading to the creation of virtual memory, enabling users to run programs larger than physical memory.
- Time-sharing systems also require a file system, disk management, and mechanisms for job synchronization and communication to ensure orderly execution and prevent deadlock.
- **Note that one processor (CPU) can only run one process (task) at a time. CPU gives small time to each process and switch to another process.**


Task 1
Task 2
Task 3
CPU gives 10 Nano seconds to task 1
CPU gives 10 Nano seconds to task 2
CPU gives 10 Nano seconds to task 3

Multi-Tasking In Operating System

As shown in the diagram above, three tasks are running on the computer. CPU gives 10 nanoseconds to each task. Time is shared between task 1, task 2 and task 3. If task 1 is not completed in 10 nanoseconds, then task 1 has to wait until task 2 and task 3 are given time.

Advantages

i. **Efficient Resource Utilization**: By allocating time slices to each task, time-sharing systems make effective use of CPU and memory, minimizing idle time.

ii. **Reduced CPU Idle Time**: Switching quickly between tasks keeps the CPU busy and improves system performance by reducing idle time.

iii. **Enhanced User Interaction**: Time-sharing allows multiple users to interact with the system simultaneously, providing immediate responses to user commands.

iv. **Improved Productivity**: Users experience reduced wait times, which leads to higher productivity as tasks are completed more promptly.

v. **Multi-User Capability**: Time-sharing allows multiple users to access a single system at the same time, making it suitable for environments like universities and businesses with shared computing needs.

vi. **Supports Multitasking**: It can handle multiple tasks at once by quickly switching between them, which improves the system's overall functionality and utility.

Disadvantages

i. **Overhead for Context Switching**: Frequent switching between tasks (context switching) can consume a significant amount of CPU time, potentially impacting performance.

ii. **Security and Data Integrity Risks**: With multiple users accessing the system simultaneously, there's a higher risk of unauthorized access, data corruption, or security breaches.

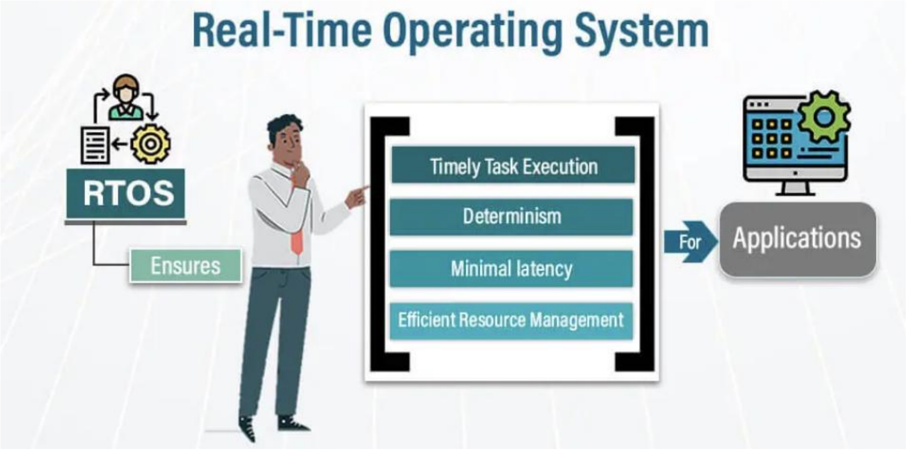
iii. **Complexity of Resource Management**: Managing resources across multiple tasks and users adds complexity, which can lead to issues if not well managed.

iv. **Reliability Concerns**: In a time-sharing system, a single user's program crash or resource overuse can impact other users, potentially affecting the whole system.

v. **Response Time Variability**: As more users or tasks are added, the system's response time for each user can fluctuate, leading to possible delays in user interaction.

4. Real Time Operating System

- A real-time operating system is an important type of operating system used to provide services and data processing resources for applications in which the time interval required to process & respond to input/output should be so small without any delay real-time system.
- For example, real-life situations governing an automatic car, traffic signal, nuclear reactor or an aircraft require an immediate response to complete tasks within a specified time delay.
- Hence, a real-time operating system must be fast and responsive for an embedded system, weapon system, robots, scientific research & experiments and various real-time objects.



Types of the real-time operating system:

Hard Real-Time System

- These types of OS are used with those required to complete critical tasks within the defined time limit.
- If the response time is high, it is not accepted by the system or may face serious issues like a system failure.
- In a hard real-time system, the secondary storage is either limited or missing, so this system stores data in the ROM.

Soft Real-Time System

- A soft real-time system is a less restrictive system that can accept software and hardware resource delays by the operating system.
- In a soft real-time system, a critical task prioritizes less important tasks, and that priority retains active until the completion of the task.
- Also, a time limit is set for a specific job, which enables short time delays for further tasks that are acceptable.
- For example, computer audio or video, virtual reality, reservation system, projects like undersea, etc.

Advantages

i. **Predictable Task Execution:** RTOS provides consistent, predictable response times, ensuring that tasks are completed within specified time constraints, which is essential for time-sensitive applications.

ii. **Reliability and Stability:** RTOS systems are designed for stability and reliability, making them suitable for mission-critical applications in fields like healthcare, aviation, and automotive.

iii. **Efficient Resource Allocation:** By prioritizing tasks and managing resources effectively, RTOS can perform complex operations with minimal delay, ensuring high efficiency.

iv. **Minimal Downtime:** RTOS are optimized to avoid unnecessary context switches or delays, reducing downtime and increasing the overall system availability.

v. **Prioritization of Tasks:** RTOS can prioritize critical tasks over less critical ones, ensuring that high-priority functions are performed immediately.

Disadvantages

i. **Complex Design and Development:** Developing an RTOS requires specialized expertise and complex coding, which increases the time and cost of development.

ii. **Limited Multitasking:** RTOS often restricts multitasking to ensure that timing constraints are met, which can be a limitation in applications needing extensive parallel processing.

iii. **Higher Costs:** Due to their complex requirements, RTOS systems are often more expensive to design, test, and implement than general-purpose systems.

iv. **Less Flexibility:** RTOS are optimized for specific tasks and are generally less adaptable to changes, which limits their use in versatile applications.

v. **Resource Intensive:** RTOS requires precise timing mechanisms and high-performance processors to ensure that deadlines are met, which may increase hardware costs and resource consumption.

5. **Mobile OS**

- A Mobile Operating System is a specialized OS designed for smartphones, tablets, and other handheld devices.
- It provides a platform for running applications, managing hardware, and handling user interactions on mobile devices.
- Popular mobile OSs include Android, iOS, Windows Mobile, and HarmonyOS.
- Mobile OSs are optimized for touch interfaces, energy efficiency, connectivity, and multimedia functionalities, making them suitable for the unique requirements of portable devices.

Key Features of Mobile Operating Systems

1. **Touchscreen Interface:** Optimized for gestures like tapping, swiping, and pinching.
2. **Connectivity:** Built-in support for Wi-Fi, Bluetooth, NFC, cellular networks, and other connectivity options.
3. **Application Ecosystem:** Access to app stores (like Google Play and Apple's App Store) for easy app downloads and installations.
4. **Security:** Sandboxing and permission models to limit app access to sensitive data and functions.
5. **Energy Efficiency:** Mobile OSs are optimized to minimize battery drain and manage power effectively.

Advantages

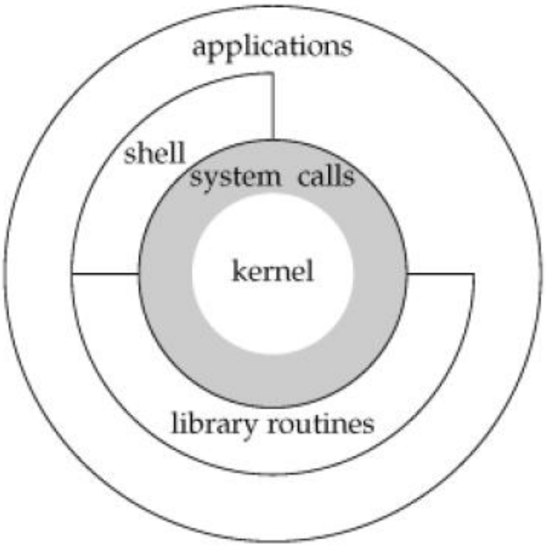
1. **User-Friendly Interface:** Mobile OSs are designed for ease of use, with simple navigation, icons, and a focus on touch-based interactions.
2. **Mobility and Portability:** The OS is optimized for mobile devices, making it convenient for users to carry, access, and use applications on the go.
3. **Vast App Ecosystem:** Mobile OSs come with access to app stores, offering users millions of apps across various categories, from productivity to entertainment.
4. **High Connectivity Options:** Mobile OSs offer seamless connectivity to cellular networks, Wi-Fi, and Bluetooth, supporting activities like communication, navigation, and media streaming.
5. **Regular Updates and Improvements:** Leading mobile OSs (like iOS and Android) provide frequent updates to enhance security, features, and performance.
6. **Enhanced Security:** Mobile OSs implement sandboxing and permission controls, ensuring each app operates within a secure environment, protecting user data.

Disadvantages

1. **Limited Multitasking:** Many mobile OSs have restricted multitasking capabilities to save battery and system resources, which can be limiting for power users.

2. **Dependency on Battery:** Mobile OSs rely on battery power, and running multiple apps or high-performance tasks can lead to quick battery drain.
3. **Platform Fragmentation:** Differences in hardware and OS versions (especially in Android) can lead to fragmentation, resulting in compatibility issues and delayed updates.
4. **Security Vulnerabilities:** Mobile devices are often targets for malware and hacking, especially due to the downloading of unverified apps or public network usage.
5. **Storage Constraints:** Mobile devices typically have less storage capacity compared to desktops, which can limit the number of apps and data a user can store.
6. **Hardware Dependence:** The performance of a mobile OS is often limited by the device's hardware (e.g., processing power, RAM), which may vary widely across different devices.

1.4 Elements of an Operating System



Kernel

- Kernel is a program, which is loaded in memory when the system is turned on.
- It stays there and provides various services until the system is turned off.
- Kernel interacts with the hardware directly.
- When a user program needs to use any hardware, it has to use services provided by the kernel. Special functions, called system calls, are used to request kernel, kernel performs the job on behalf of the user process.
- In addition to providing services to user programs, the kernel also provides other services like process management, memory management, file system management so on.
- In short, the kernel manages the entire computer system.

Shell

- The shell is an interface between the user program and the kernel.

- When the user logs in to the system, the process for the shell starts execution.
- It terminates when the user logs out from the system.
- Users can directly interact with the shell.
- It works as a command interpreter.
- It accepts commands from a user and translates them into a form, which the kernel can understand easily.
- It is also a programming language.
- It provides various programming functionalities such as looping, branching, and so on.

System Call:

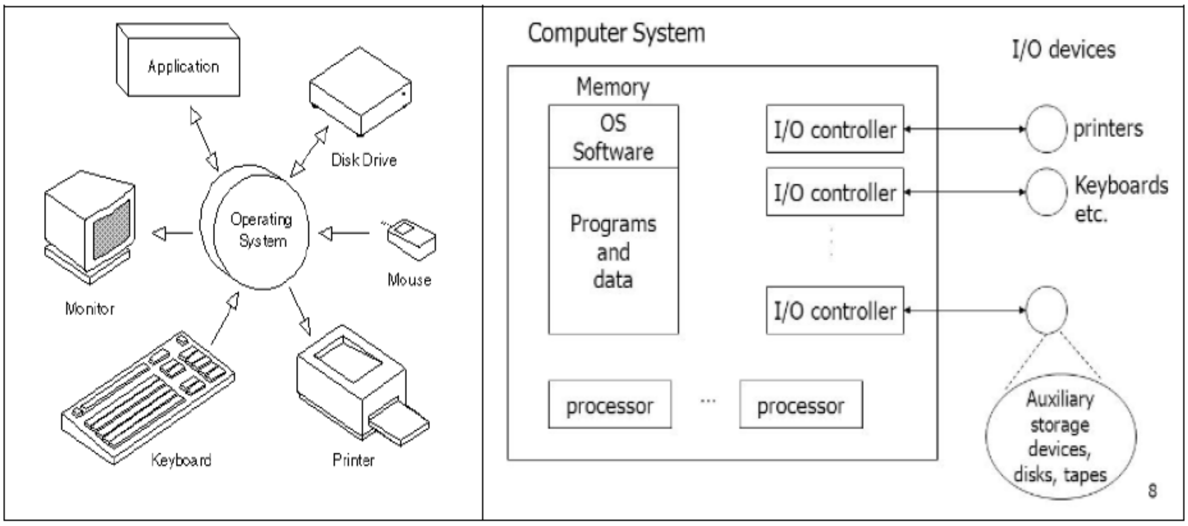
- System calls are special functions.
- They are used to request kernel to provide various services, such as reading from a file stored on hard disk.
- They can be invoked via library procedures, or via commands provided by shell, or even directly from C programs in UNIX.
- System calls are similar to user-defined functions. Difference is that they execute in the kernel mode, having fully access to all the hardware; while user defined functions execute in user mode, having no direct access to the hardware.

System call process

- When user executes function then library procedure executes TRAP to change the User mode to Kernel mode.
- Library procedure issues respective system calls.
- System calls executes within kernel, can directly deal with hardware and generates output.
- Output is given to the Library procedure.
- Again TRAP is used to change the Kernel mode to User mode.
- Library procedure gives the respective output to the user.

1.5 Operating System as a Resource Manager

- Computer System is the collection of multiple resources like Processor, input devices, Output Devices, storage devices etc.
- Internally the resources are also not able to work with each other. For this they need instructions which are given by OS.
- OS is responsible for Data Processing, Data Movement and Data Store.
- Following figures shown this concept.



- When a user wants to access any program or application, then using GUI or CUI interface, he can execute the program.
- The operating system manages the resources like:

  Storage Device
  - OS manages the data communication between storage devices.
  - It gives respective instructions to transfer the data from Secondary storage devices to main memory for execution.
  - After the completion of the task, it removes the data from the main memory and stores this information in its original place on a secondary device.

  Processor
  - OS is responsible for assigning job to the processor.
  - It takes care of only one process that remains there with the processor for execution.
  - It handles the process scheduling techniques based on which it assigns jobs to the CPU.
  - It tries to take maximum utilization of the CPU.
  - Processor executes only those processes which are assigned by the OS. If the OS doesn’t assign any process to the CPU then it remains free for that time.

  Input Devices
  - During program execution, when needs to take input from the user, the OS instructs the related Input Device to take input from the user.
  - When the user gives inputs, instantly OS gives related instructions and transfers the respective data to the computer system for further processing.

  Output Devices
  - During program execution, when system needs to give output to the user then OS instructs the output devices to display results.
  - OS sends the require information to the output device for printing.

- It also maintains the print queue in which multiple results are stored for printing in First In First Out based.