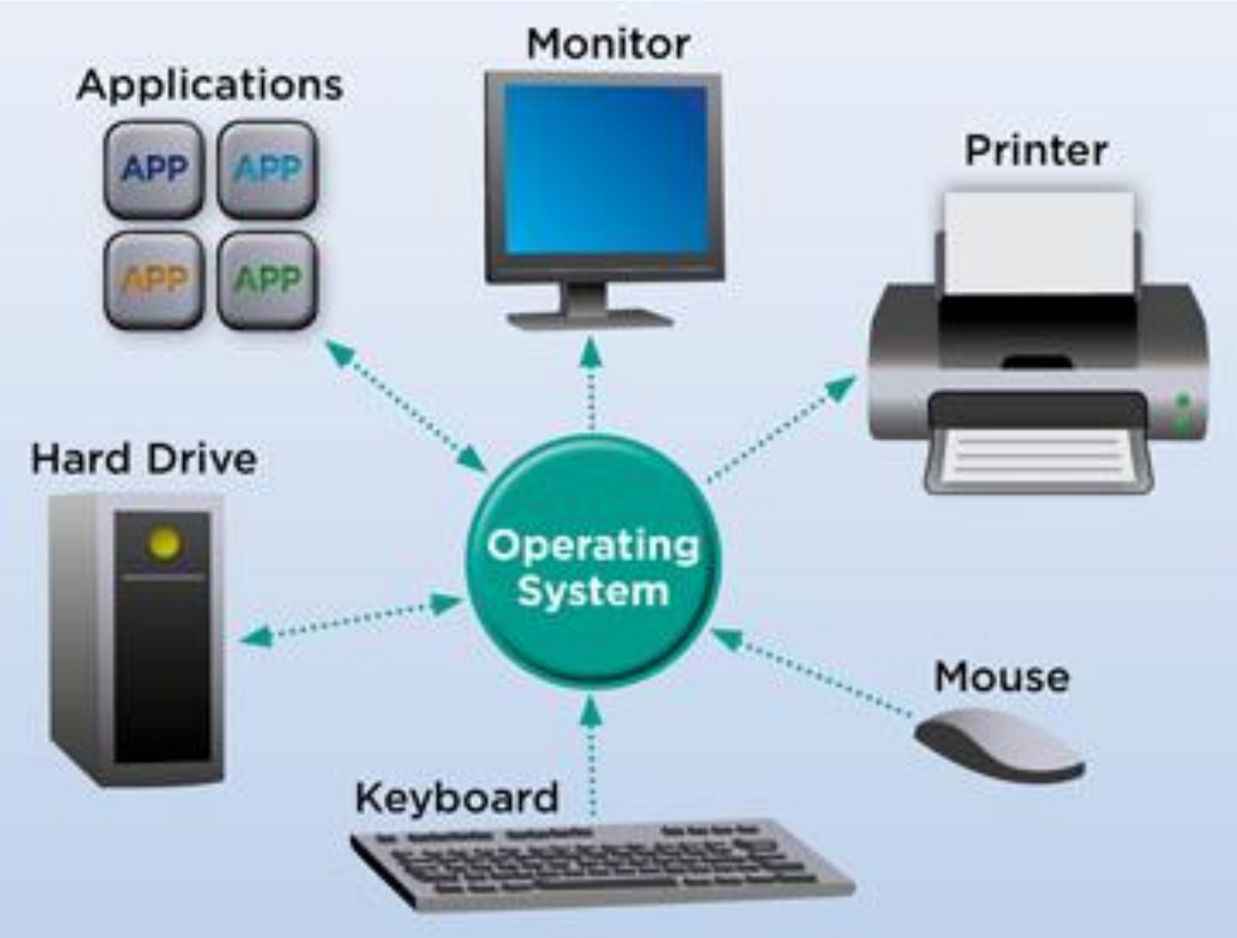
Operating Systems

UNIT-1: Part 1

Topics

- ❖ Introduction to Operating Systems
  - ○ Computer System Overview
  - ○ Components of a Computer System
  - ○ Functions of OS
  - ○ Different Types of OS
  - ○ OS Distributions and Versions

What is an Operating System?

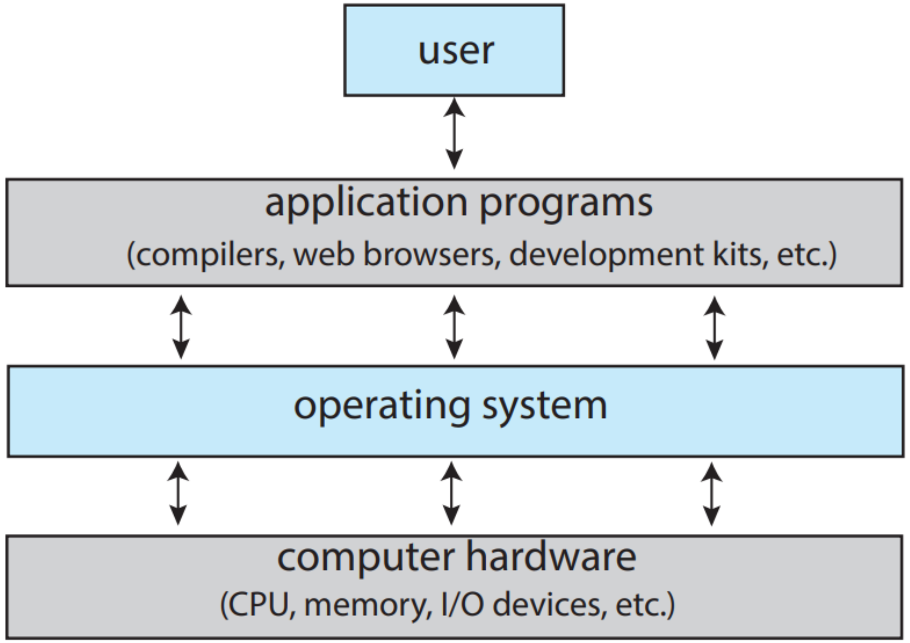


What is an Operating System?

- A program that acts as an intermediary between a user of a computer and the computer hardware
- Operating system goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

Components of a Computer System

A computer system can be divided into four components:


user
application programs
(compilers, web browsers, development kits, etc.)
operating system
computer hardware
(CPU, memory, I/O devices, etc.)

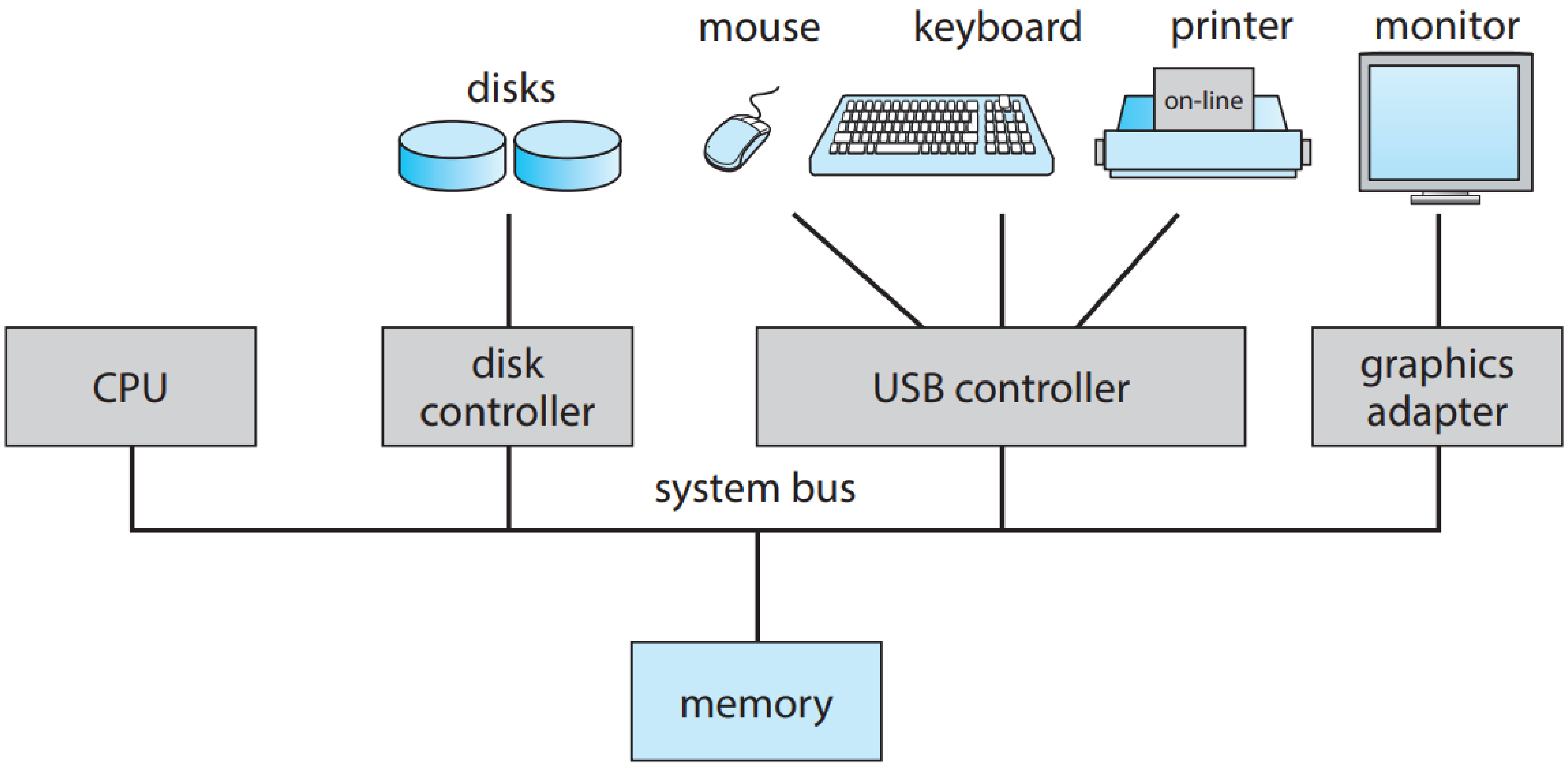
Hardware— Provides the basic computing resources the central processing unit (CPU), the memory, and the input/output (I/O) devices

Operating System — Software that controls and Coordinates use of hardware among various applications and users

Application programs—such as word processors, spreadsheets, compilers, and web browsers

Users — People, Machines and Other Computers

Computer System Organization



Why Study Operating Systems?

- Only a small percentage of Computer Science professional will be involved in the creation or modification of an operating system.
- All code runs on top of an operating system.
- Knowledge of
  - How OS works,
  - How OS drives computer hardware,
  - What OS provides to applications

  is crucial to write efficient, effective, and secure Code.

Operating System Definition

- The operating system is the one program running at all times on the computer (Kernel of the Operating System) , with all else being either
  - System programs or
  - application programs.
- OS is resource allocator
  - Allocates all the resources such as memory, processors, devices, and information.
- OS is a Control Program-includes programs to manage the resources, such as a traffic controller, a scheduler, a memory management module, I/O programs, and a file system.

Functions of Operating System (1)

Resource Management	**M**anages and allocates memory, CPU time, and other hardware resources among programs and processes.
Memory Management	Manages the computer’s primary memory and provides mechanisms for optimizing memory usage. (Maintains memory logs)
File Management	OS is responsible for organizing and managing the file system, (Creation, deletion, and manipulation of files and directories)-NTFS, ext4
Device Management	Manages input/output devices, Maintains the status of all devices connected to the system (device log),provides the necessary drivers and interfaces
Process Management	Starting, stopping, managing and scheduling of processes.
	Allocation and deallocation of resources

Functions of Operating System (2)

Security	Provides a secure environment for the user, applications, and data (access controls and encryption)
Job Accounting	Maintains log for all users
	Keeps track of time and resources used by various jobs or users.
Error Detection	These contain methods that include the production of dumps, traces, error messages, and other debugging and error-detecting methods.
Backup and Recovery	Provides mechanisms for backing up data and recovering it in case of system failures, errors, or disasters.
Updates and Maintenance	Operating systems regularly receive updates and patches to fix security vulnerabilities and improve functionality. For instance, Windows Update and Linux package managers like apt or yum.

Functions of Operating System (3)

User Interface	Provides a user interface that enables users to interact with the computer system (GUI/CLI/ both).
Networking	Provides capabilities establishing and managing network connections, network protocols,, sharing resources over a network.
System Calls	Enable apps to interact with the OS and access its resources, Provides portability and compatibility across different H/W and S/W platforms.
Time Sharing	Enables multiple users to share a computer system and its resources simultaneously (Time sharing mechanisms)
Virtualization	Virtualization capabilities allow multiple operating systems or applications to run on a single physical machine.
Performance Monitoring	Monitoring and optimizing system performance (Identifying bottlenecks, optimizing resource usage, and analyzing system logs and metrics)

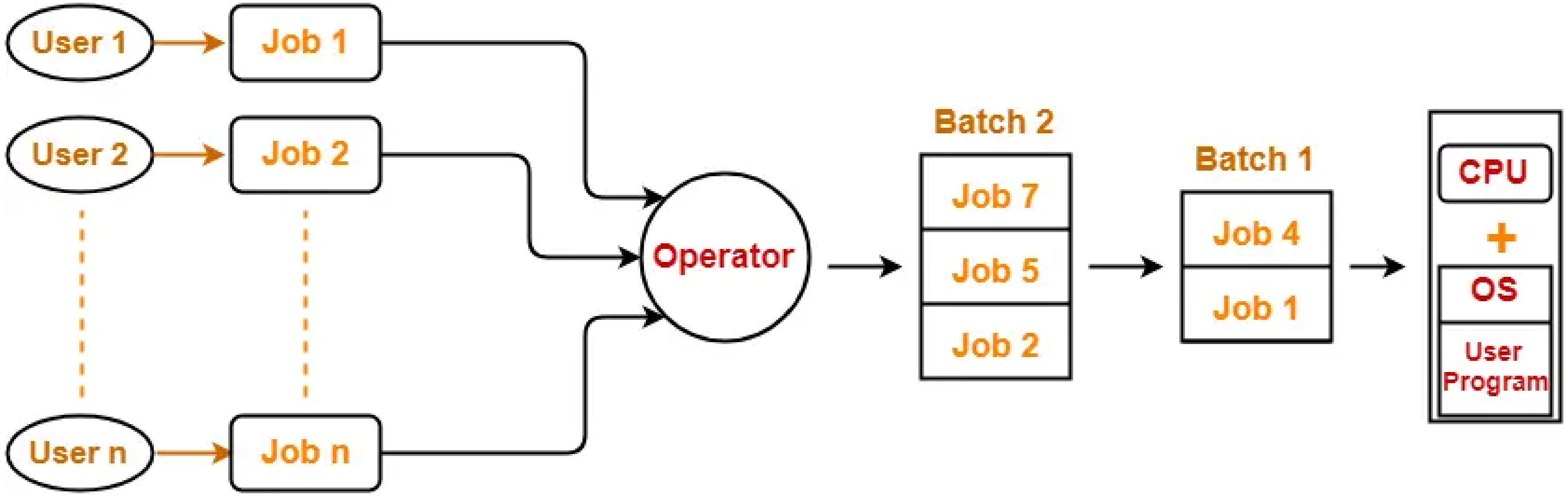
Types of Operating System

Different types of operating systems are mentioned below:

1. Batch Operating System
2. Multi-Programming System
3. Multi-Processing System
4. Multi-Tasking Operating System
5. Time-Sharing Operating System
6. Distributed Operating System
7. Network Operating System
8. Real-Time Operating System

1. Batch Operating System

- In a batch processing system, users submit their jobs to the system, and the operating system collects these jobs into batches.
- The jobs in each batch are executed one after the other without any user intervention until all the jobs in the batch have completed.



Batch Operating System

1. Batch Operating System (Contd..)

Key Characteristics of Batch processing system:

Minimal User Interaction

- Users submit their jobs to the system and receive the results when the jobs are completed.

Efficiency

- Efficient for tasks that are repetitive, time-consuming and resource intensive.
- It speeds up the process by combining similar types of jobs and runs them as a group.

Require an Operator

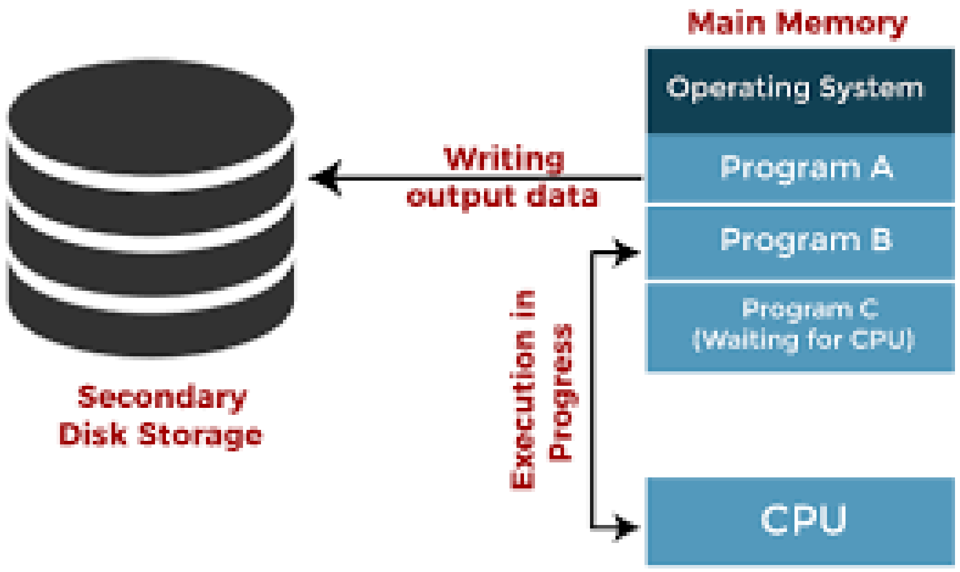
- To club similar jobs having the same requirements.

Examples

- Early Mainframe computers for applications such as Payroll processing and scientific simulations
- Still used for Large Scale data processing

2. Multi-Programming Operating System

- Before the concept of Multiprogramming, CPU executes only one program at a time.
- When the program undergoes in waiting state for an input/output operation, the CPU remains idle which leads to underutilization of CPU and thus poor performance.
- The concept of multiprogramming solves this issue by allowing more than one program to execute in an operating system



2. Multi-Programming Operating System

Features of Multiprogramming:

- Need Single CPU for implementation.
- Context switch between process.
- Switching happens when current process undergoes waiting state.

Advantages:

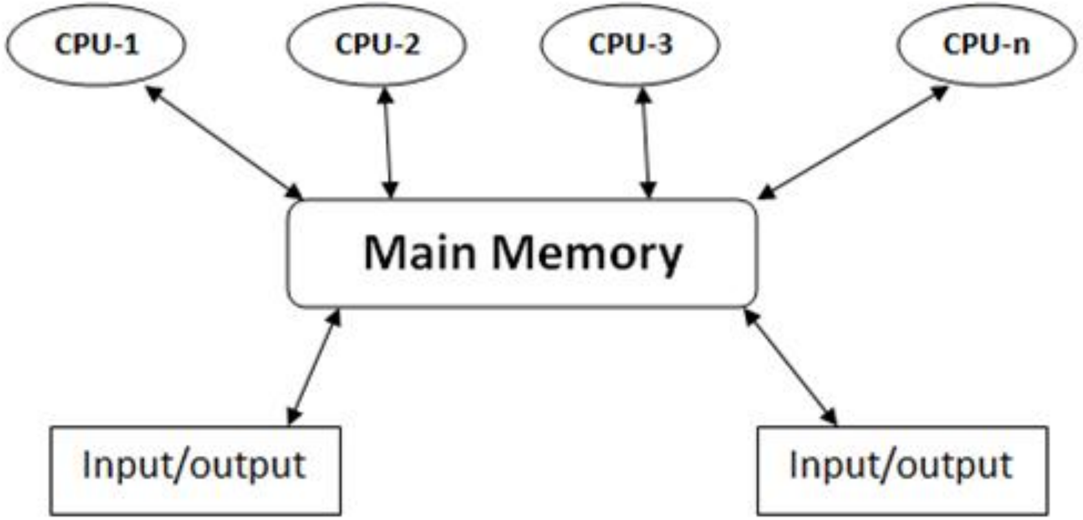
- CPU idle time is reduced.
- High resource utilization.
- High Performance.

Drawbacks:

- CPU scheduling algorithms is required.
- In case of large number of jobs, long-term jobs require a longer wait.
- Memory management is needed to meet the storage requirements of multiple programs

3. Multi-Processing Operating System

- Multiprocessing systems are computer systems that utilize multiple processors or CPU cores to execute tasks concurrently.



3. Multi-Processing Operating System

Advantages of Multi-Processing Operating System

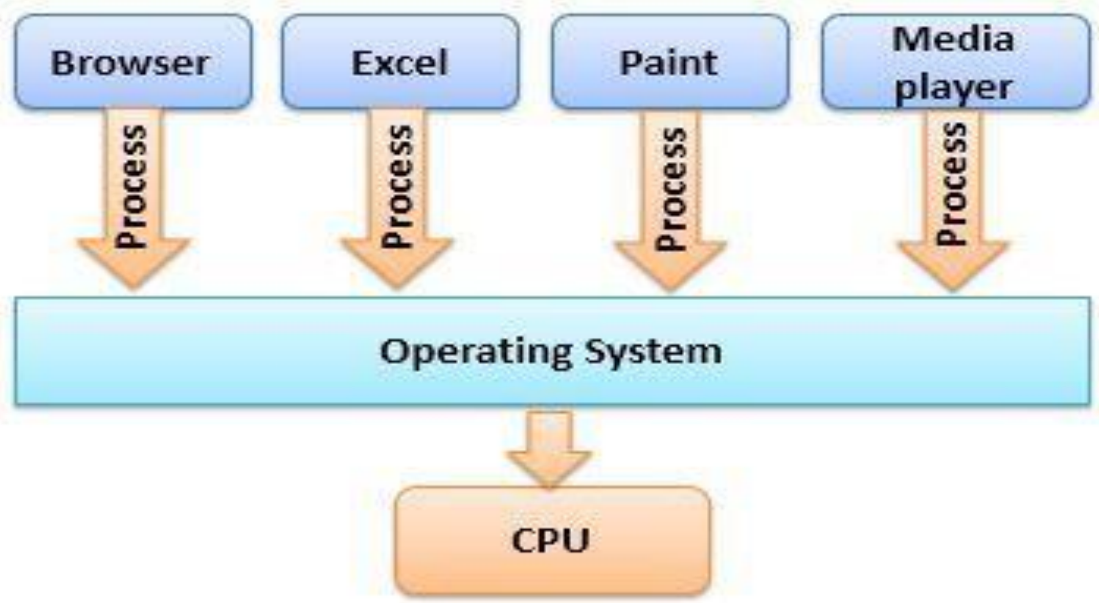
- It increases the throughput of the system.
- As it has several processors, so, if one processor fails, we can proceed with another processor.

Disadvantages of Multi-Processing Operating System

- Due to the multiple CPU, it can be more complex

4. Multi-Tasking Operating System

- Allows multiple tasks (processes) to run concurrently on a single computer system.
- CPU switches rapidly between different tasks, giving the appearance that multiple tasks are running simultaneously.



Multitasking

4. Multi-Tasking Operating System

Advantages:

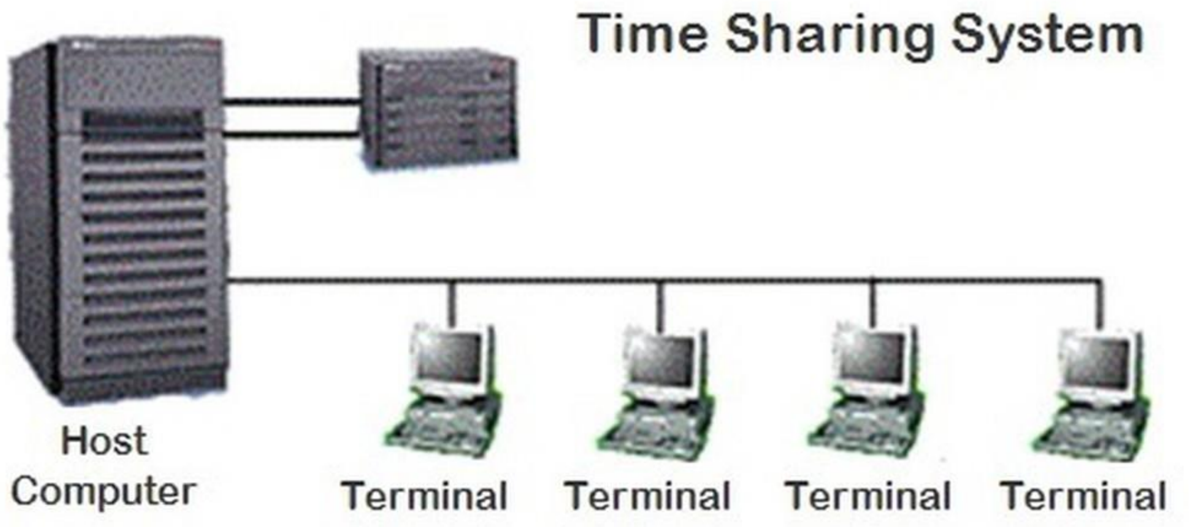
- ➢ allowing users to work on multiple tasks simultaneously, reducing downtime and improving efficiency.
- ➢ Maximize resource utilization such as CPU time and memory

Drawbacks:

- ➢ Managing multiple tasks concurrently adds complexity to the operating system.
- ➢ Resource Contention can occur when multiple tasks compete for the same resources

5. Time Sharing Operating System

- A time-sharing operating system enables concurrent use of the computer by multiple users.
- Processor's time is shared among multiple users simultaneously



5. Time Sharing Operating System

Advantages

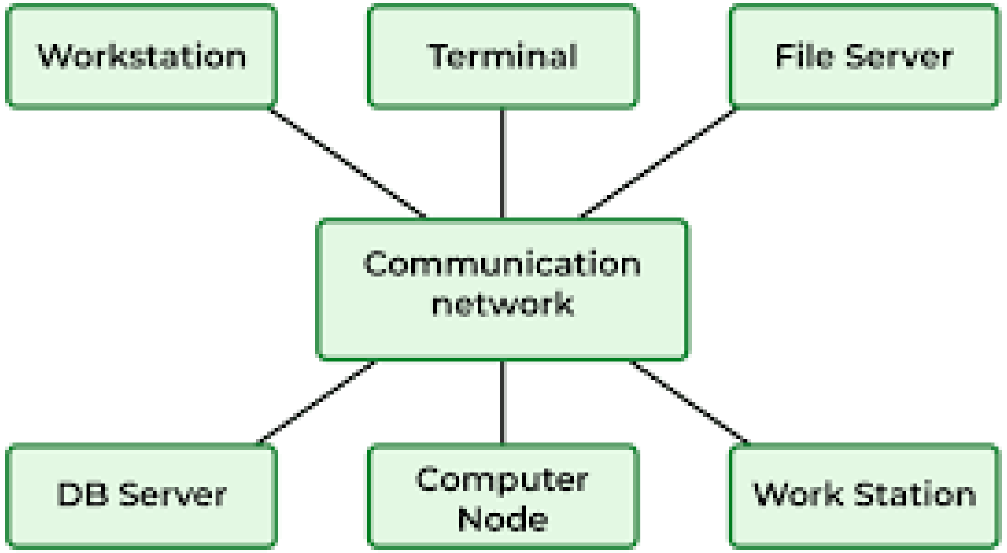
- Enables many users located at different terminals to simultaneously use a single computer system.
- Avoids duplication of software.
- Reduces idle time of CPU

Disadvantages/ Challenges:

- Communication Overhead
- Ensuring Reliability
- Preserving Security and Integrity of user programs

6. Distributed Operating System

- Various autonomous interconnected computers communicate with each other using a shared communication network.
- Independent systems possess their own memory unit and CPU.
- Remote access is enabled within the devices connected in that network.



6. Distributed Operating System

Advantages

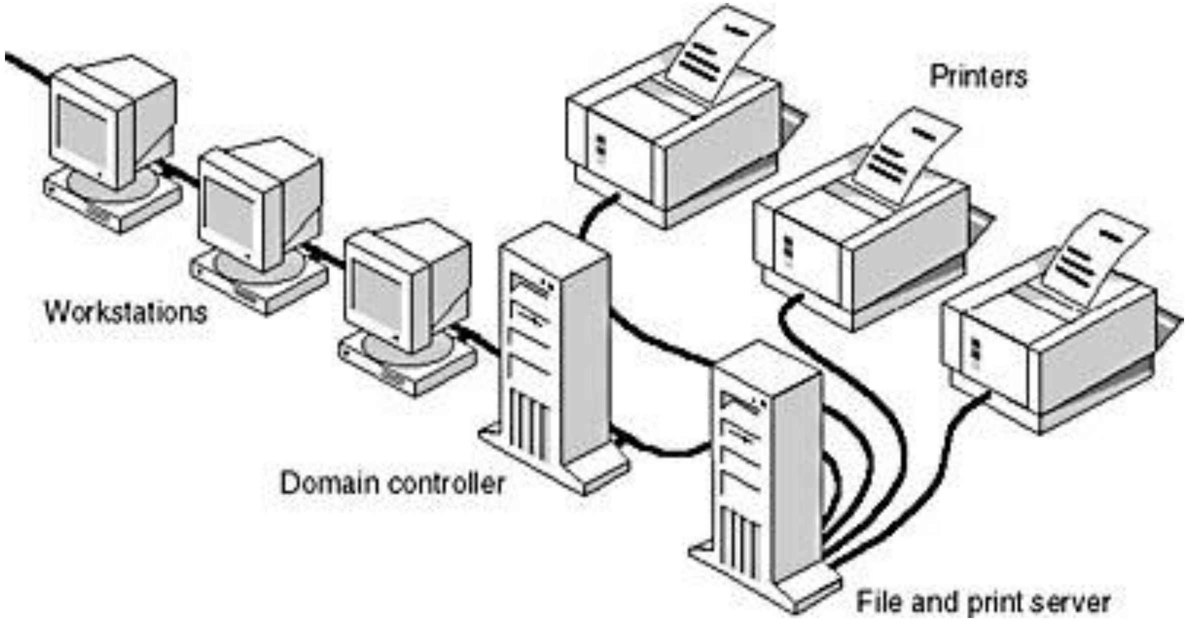
- Allow remote working.
- Faster exchange of data among users.
- Failure in one site may not cause much disruption to the system.
- Minimize the load on the host computer.

Disadvantages

- If the primary network fails, entire system shuts down.
- Expensive to install.
- Require a high level of expertise to maintain.

7. Networked Operating System

- Networked operating systems primarily focus on managing network-related operations and services within a single computer or node.
- They are designed to facilitate communication and connectivity within the local network.
- Eg: Microsoft Windows Server 2003, Microsoft Windows Server 2008, Linux and Mac OS X.



7. Networked Operating System

Advantages:

- Centralized servers are highly stable.
- Security is server managed.
- Upgradation of new technologies and hardware can be easily integrated into the system.
- It is possible to remotely access servers from different locations

Disadvantages:

- High cost of buying and running a server.
- Dependency on a central location for most operations.
- Regular maintenance and updates are required.

8. Real-Time Operating System

- Real-time **operating systems (RTOS)** are used in environments where a large number of events must be accepted and processed in a short time or within certain deadlines.

Hard real-time systems	Guarantee that critical tasks complete on time. Secondary storage is limited or missing. Examples: Medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems etc.
Soft real-time systems	Soft real-time systems are less restrictive. Soft real-time systems have limited utility than hard real-time systems. Examples: Multimedia, virtual reality, Advanced Scientific Projects like undersea exploration etc.

8. Realtime Operating System

Advantages:

- Maximum utilization of devices and systems.
- Time assigned for shifting tasks in these systems is very less.
- Since the size of programs is small, RTOS can also be embedded systems
- These types of systems are error-free.

Disadvantages:

- Very few tasks run simultaneously, and their concentration is very less on few applications to avoid errors.
- The algorithms are very complex and difficult for the designer to write on.
- It needs specific device drivers and interrupts signals to respond earliest to interrupts.
- RTOS performs minimal task switching.

Operating System Distributions

- An operating system distribution is a complete package of an operating system that includes the core OS kernel, system utilities, libraries, and often a collection of software applications and tools.
- These distributions are typically customized and bundled to serve various purposes and cater to different user needs.
- Some well-known operating system distributions include:

Linux Distributions:

- Linux distributions like Ubuntu, Debian, CentOS, Fedora, and Arch Linux are examples of operating system distributions.
- Cater to desktop, server, and embedded use cases.

Operating System Distributions

UNIX Variants: UNIX-like operating systems such as macOS, AIX (Avanced Interactive Executive), HP-UX (Hewlettte Packard Unix), and Solaris have their own distributions or versions customized for specific hardware and enterprise environments.

Windows Editions: Microsoft Windows has multiple editions like Windows 10 Home, Windows 10 Pro, Windows Server, and Windows Enterprise, each tailored for different user categories and needs.

Operating System Versions

- Operating systems are updated and improved over time to fix bugs, enhance security, and add new features.
- Each new version typically receives a unique version number or name.
- Operating system versions are crucial because they indicate the level of updates, security patches, and new features available in a particular release.
- Users and administrators can choose the version that best suits their needs based on factors like stability, support, and compatibility with their hardware and software requirements.

Operating System Versions

Windows 1
Windows 3.1
Windows 95
Windows XP
Windows Vista
Windows 7
Windows 8
Windows 10
Windows 11
Windows 12

Quiz Time

1. Which OS reads and reacts in terms of actual time?
2. A systematic procedure for moving the CPU to new process is known as ____________
3. UNIX is written in which language?
4. Main memory of a computer system is? (Volatile/Non-Volatile)
5. The speed of writing data in magnetic tape disks is comparable to that of disk drives. State True/False

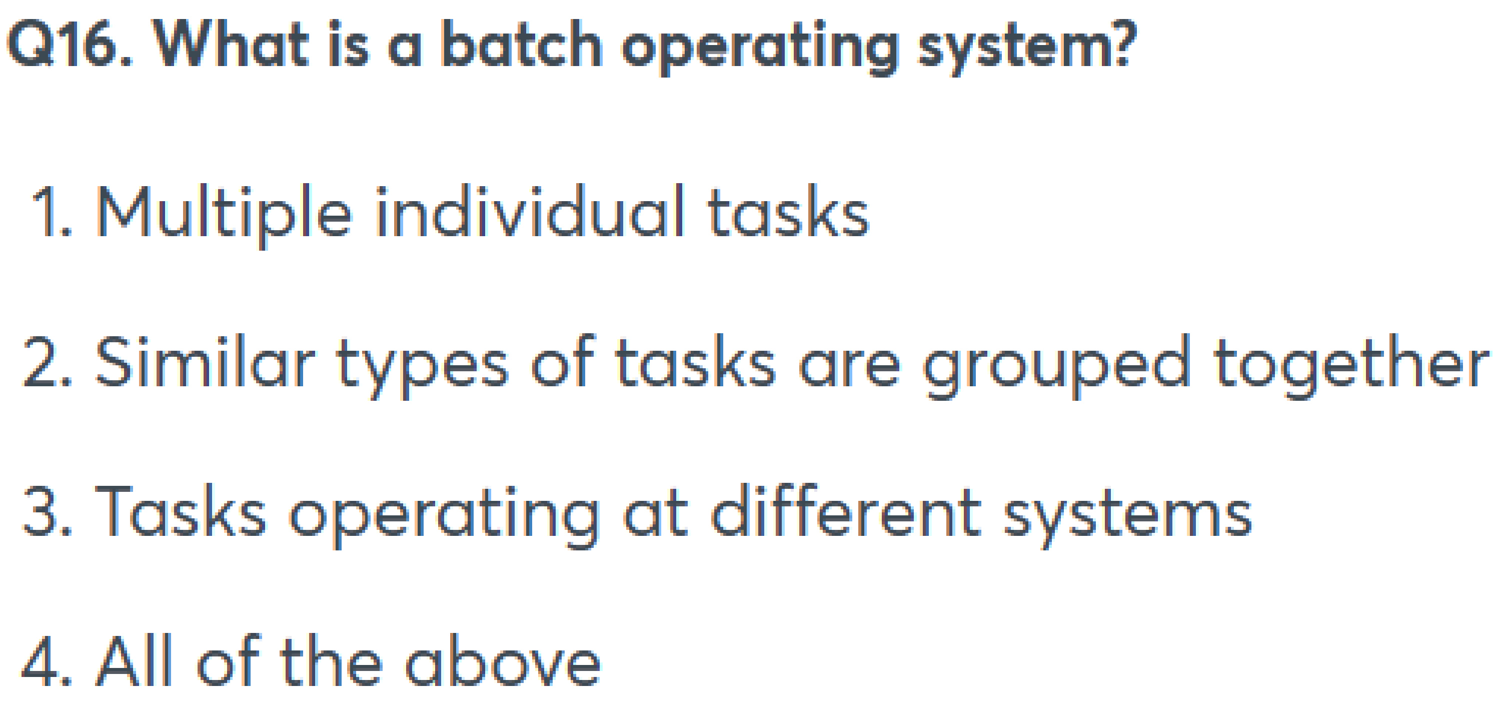
Quiz Time

6. Which of the following is not an operating system?

Linux/Dos/Oracle/Windows

7. What type of extension name do notepads use?
8. Where are the errors and bugs recorded? (Notepad/New program/ Running process/Logfile)

Quiz Time (Q9)



Q16. What is a batch operating system?

1. Multiple individual tasks
2. Similar types of tasks are grouped together
3. Tasks operating at different systems
4. All of the above

Thank you

Dr. K. Radhika

Professor, AI&DS Department

Mail:kradhika_aids@cbit.ac.in

Contact No.: 9346253642

Operating Systems

UNIT-1: Part 2

OS Architectures

OS Architecture: Introduction

- Operating system architectures can be categorized into several different types based on their design and structure.

- OS architectures determine how the operating system interacts with hardware, manages processes, and provides services to user programs.

OS Architecture:

- The architecture of an operating system consists of four major components:
  - hardware
  - kernel
  - shell
  - application

OS Architecture:

- Application: The software that a user is running on an operating system it can be either system or application software.

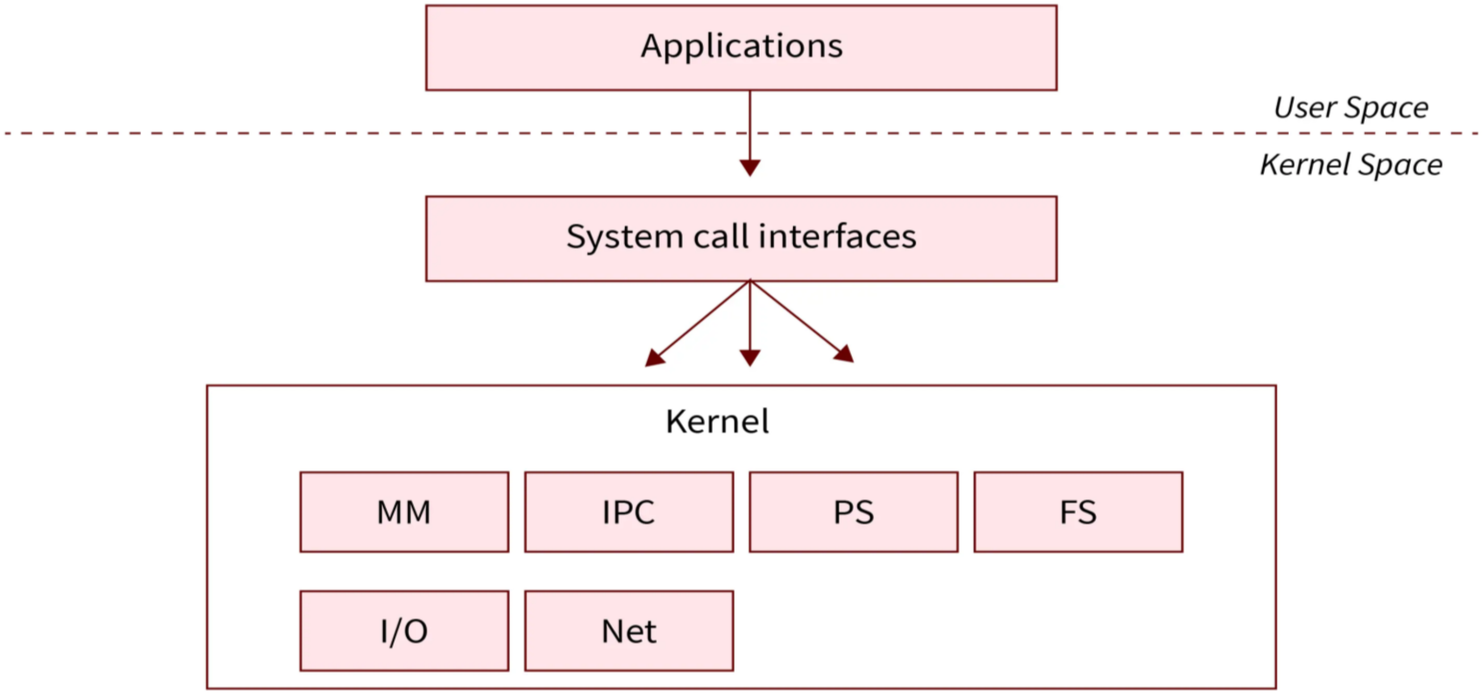
- Shell: The shell represents software that provides an interface for the user where it serves to launch or start some program for which the user gives instructions.

- Kernel: Kernel represents the most central and crucial part of the operating system:
  - Resource management
  - Provides necessary I/O, processor, and memory to the application processes
  - Inter-Process Communication, Message Passing

Types of OS Architectures:

- Architectures of operating systems can be of four types:
  - Monolithic
  - Layered
  - Microkernel, and
  - Hybrid.

Monolithic Architecture:



Applications

User Space

Kernel Space

System call interfaces

Kernel

MM

IPC

PS

FS

I/O

Net

MM - Memory Scheduler
IPC - Interprocess Communicatiom
PS - Process Scheduler
FS - File System
I/O - Input / Output Manager
Net - Network Manager

Monolithic Architecture:

- In monolithic architecture, each component of the operating system is contained in the kernel i.e. it is working in kernel space
- The components of the operating system communicate with each other using function calls.
- Examples: OS/360 (IBM-1960's) VMX (Virtual Machine eXtensions by Intel) , and LINUX

Monolithic Architectures: Advantages

- In a single address space, the entire large process is running.
- Provides CPU scheduling, memory management, device management, etc. through system calls.
- It is a single static binary file.
- Simple and easy to implement
- Faster execution due to direct access to all the services

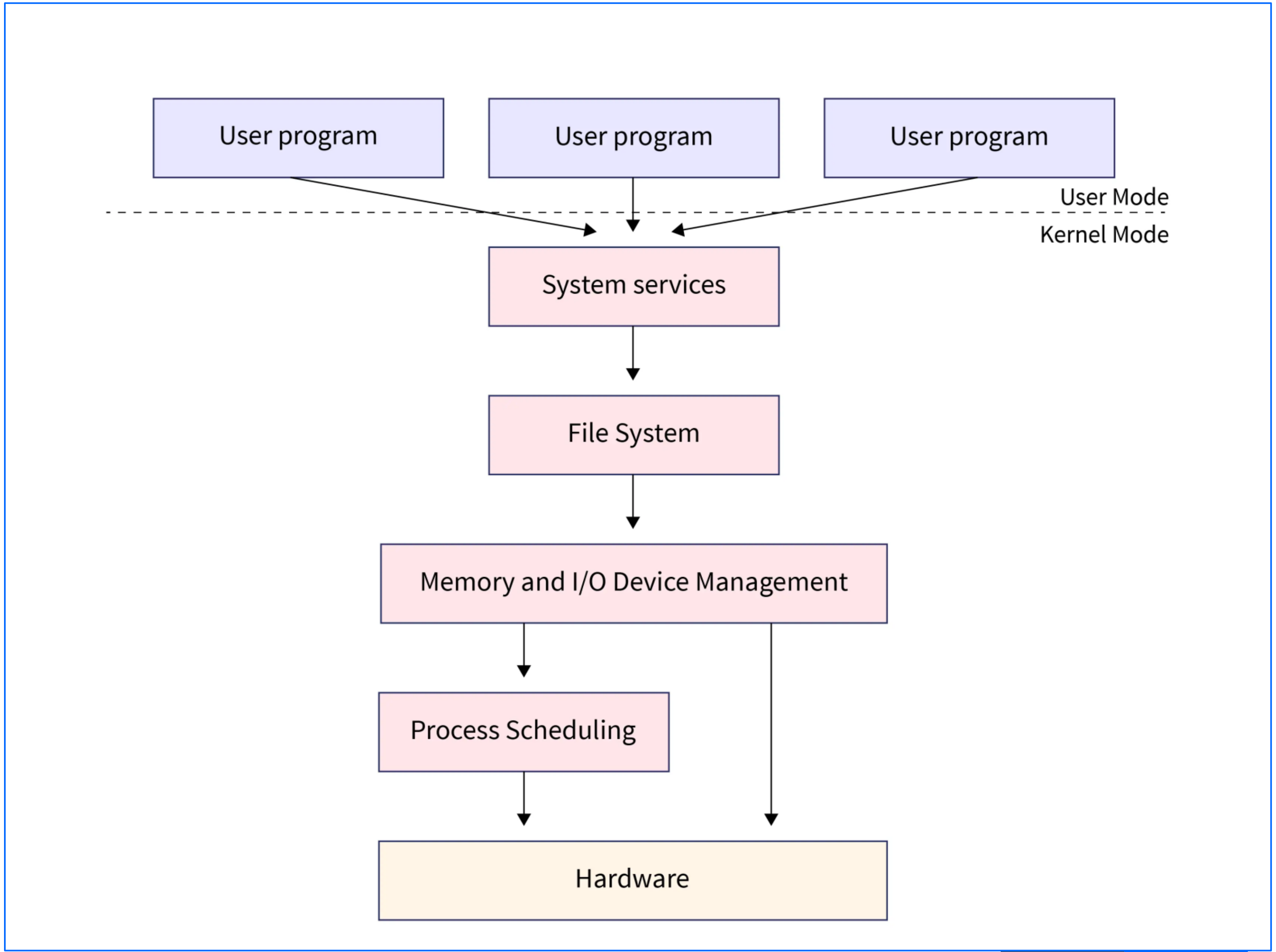
Monolithic Architectures: Drawbacks

- Addition of new features or removal of obsolete features is very difficult.
- All components are inter-dependent and when one of them fails the entire system fails.
- Security issues are always there because there is no isolation among various components present in the kernel.

Layered Architectures:

- This is an important architecture of operating system which is meant to overcome the disadvantages of early monolithic systems.
- In this approach, OS is split into various layers such that all the layers perform different functionalities.

Layered Architectures:



Layered Architectures:

- In Layered architecture, components with similar functionalities are grouped to form a layer.
- A total n+1 layers are constructed and counted from 0 to n where each layer has a different set of functionalities and services.
- Example: The operating systems windows XP, and LINUX implements some level of layering.
- The layers are implemented according to the following rules:
  - Each layer can communicate with all of its lower layers but not with its upper layer i.e. any ith layer can communicate with all layers from 0 to i-1 but not with the i+1th layer.
  - Each layer is designed in such a way that it will only need the functionalities which are present in itself or the layers below it.

Layered Architectures:

1. Hardware:

This layer is the lowest layer in the layered operating system architecture, this layer is responsible for the coordination with peripheral devices such as keyboards, mice, scanners etc.

2. CPU scheduling:

- ✓ This layer is responsible for process scheduling, multiple queues are used for scheduling.
- ✓ Process entering the system are kept in the job queue while those which are ready to be executed are put into the ready queue.

Layered Architectures: Advantages

3. Memory Management: This layer handles the aspect of memory management i.e. moving the processes from the secondary to primary memory for execution and vice-versa.

There are memories like RAM and ROM. RAM is the memory where our processes run. They are moved to the RAM for execution and when they exit they are removed from RAM.

4. Process Management: This layer is responsible for managing the various processes i.e. assigning the CPU to those processes on a priority basis for their execution. Process management uses many scheduling algorithms for prioritizing the processes for execution such as the Round-Robin algorithm, FCFS(First Come First Serve), SJF(Shortest Job First), etc.

Layered Architectures: Advantages

5. I/O Buffer: Buffering is the temporary storage of data and I/O Buffer means that the data input is first buffered before storing it in the secondary memory. All I/O devices have buffers attached to them for the temporary storage of the input data because it cannot be stored directly in the secondary storage as the speed of the I/O devices is slow as compared to the processor.

6. User Programs: This is the application layer of the layered architecture of the operating system, it deals with all the application programs running eg games, browsers, words, etc. It is the highest layer of layered architecture.

Layered Architectures: Advantages

1. Layered architecture of the operating system provides modularity
2. Easier testing and debugging due to isolation among the layers.
3. Changing or updating one of OS components will not affect the other component (easy to debug and update)
4. Dysfunction of one layer will not affect the entire operating system
5. Adding new functionalities or removing the obsolete ones is very easy.
6. The user can access the services of the hardware layer but cannot access the hardware layer itself because it is the innermost layer. (Security)

Layered Architectures: Drawbacks

1. Layered architecture is complex in implementation because one layer may use the services of the other layer and therefore, the layer using the services of another layer must be put below the other one. (Design Complexity)

2. In a layered architecture, if one layer wants to communicate with another it has to send a request which goes through all layers in between which increases response time causing inefficiency in the system. (Delays)

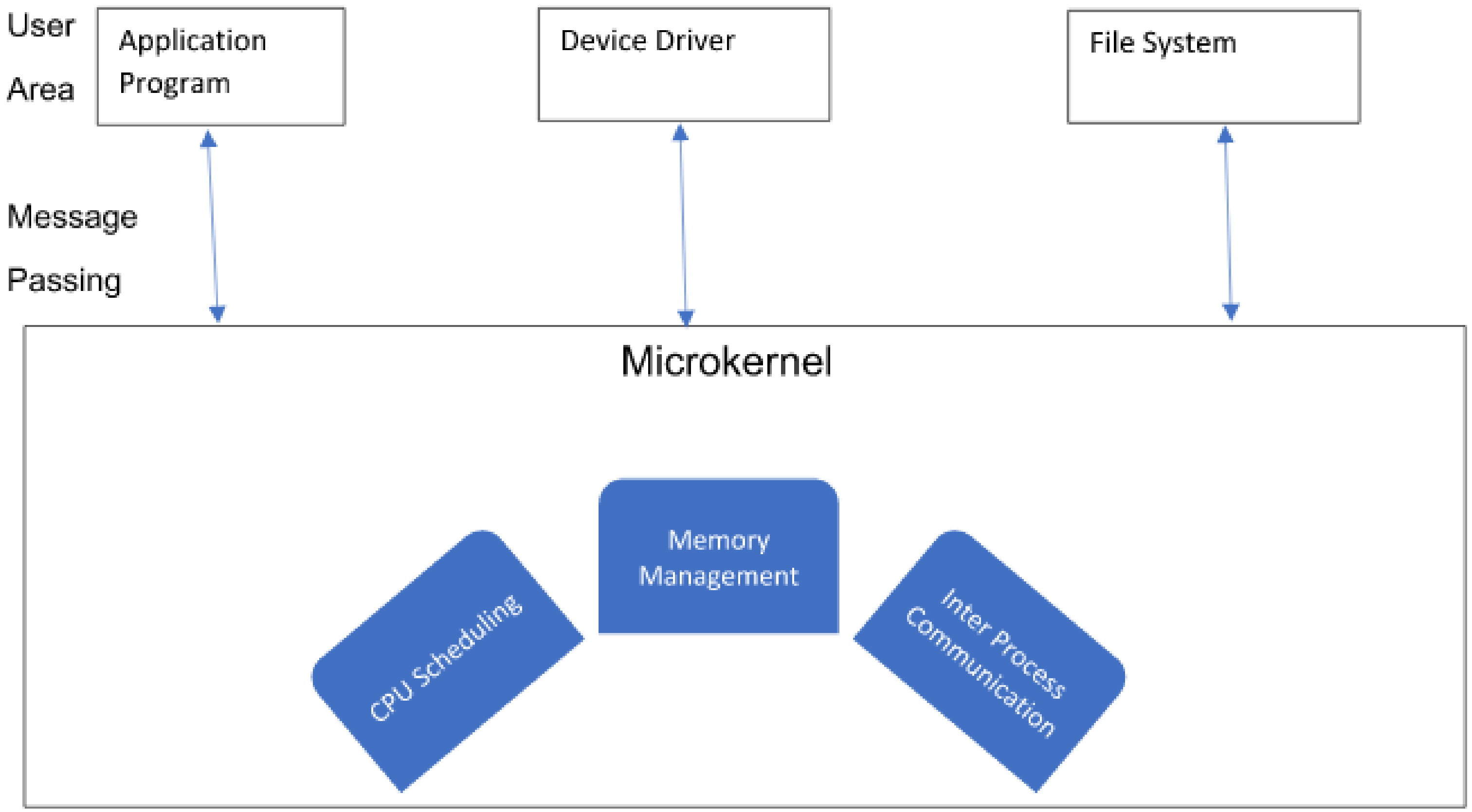
Micro-Kernel Architecture

- The basic ideology in this architecture is to keep the kernel as small as possible.
- Kernel is meant for handling the most important services only.
- In microkernel architecture, only the most important services are put inside the kernel and rest of the OS service are present in the system application program.
- Microkernel is solely responsible for the three most important services of operating system namely:
  - Inter-Process communication
  - Memory management
  - CPU scheduling

Micro-Kernel Architecture

- In a microkernel architecture, the core of the operating system is minimal, containing only essential services such as process management, memory management, and inter-process communication.
- Additional services like file systems and device drivers run as separate user-space processes rather than within the kernel.
- The processes inside the kernel have relatively high priority
- The components possess high modularity hence even if one or more components fail the operating system keeps on working.
- Eg: WindowsXP, Linux

Micro-Kernel Architecture



Micro-Kernel Architecture: Advantages

1. Microkernel operating systems are modular. Hence, disturbing one of the components will not affect the other component.
2. The architecture is compact and isolated and hence relatively efficient.
3. New features can be added without recompilation.

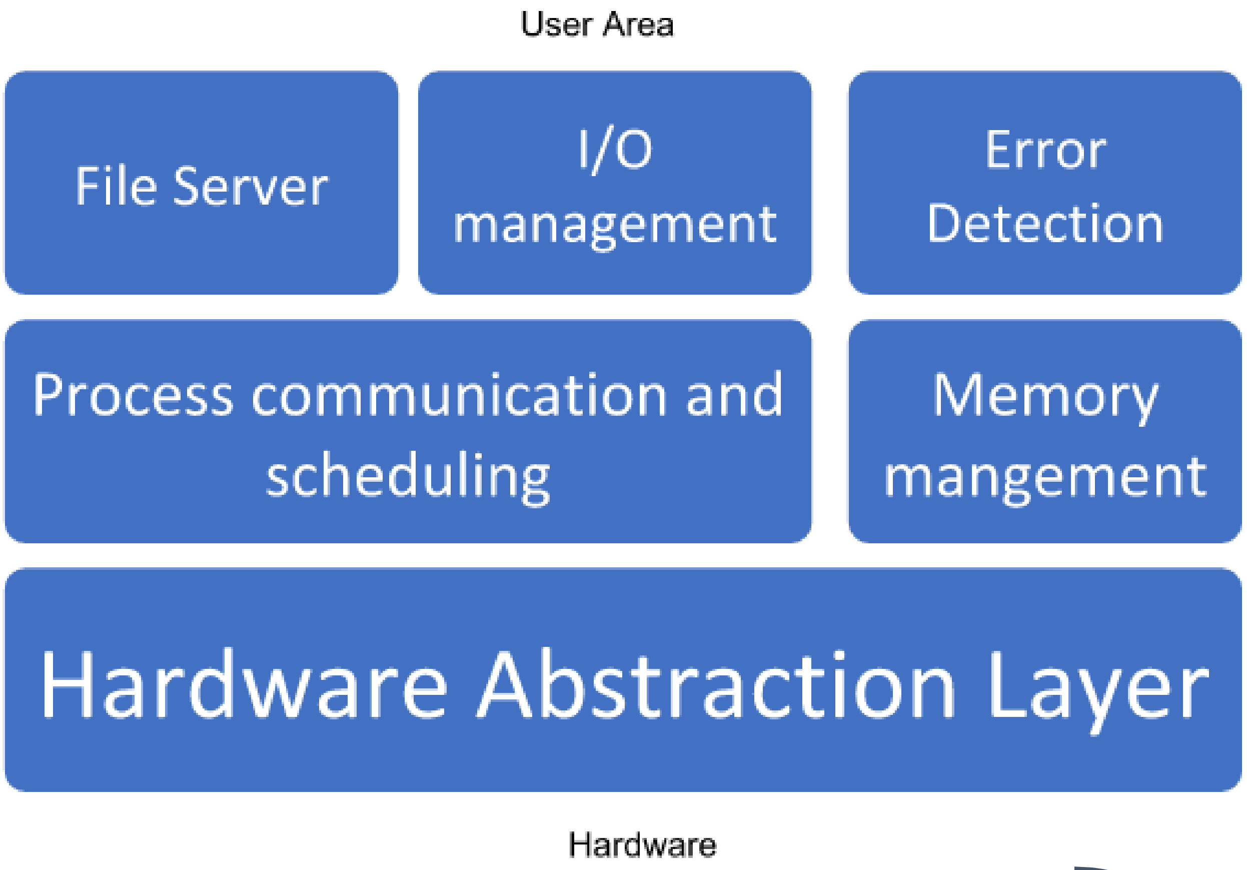
Micro-Kernel Architecture: Drawbacks

1. Microkernel architectures aim to reduce kernel complexity by moving services to user space, but they introduce complexity in managing multiple user-level components.
2. Because services and drivers run in user space rather than kernel space, communication between them often involves context switches and inter-process communication (IPC).
3. Real-time and high-performance applications may not be well-suited for microkernel-based systems

Hybrid Architecture

- Monolithic systems are quite fast but their expansion is very difficult.
- Layered structure gives an efficient division of functionalities but if the number of layers is very high, it is difficult to manage the system.
- Microkernel architecture is quite efficient in isolating the core functionalities within the microkernel but the other services which are outside the kernel are not properly integrated.
- Hybrid architecture combines the best functionalities of all these approaches.

Hybrid Architecture



User Area

File Server

I/O management

Error Detection

Process communication and scheduling

Memory mangement

Hardware Abstraction Layer

Hardware

Hybrid Architecture

The hybrid-architecture consists of three layers

1) Hardware abstraction layer: It is the interface between the kernel and hardware and is present at the lowest level.

2) Microkernel Layer: This is the old microkernel that we know and it consists of CPU scheduling, memory management, and inter-process communication.

3) Application Layer: It acts as an interface between the user and the microkernel. It contains the functionalities like a file server, error detection, I/O device management, etc.

Example: Microsoft Windows NT kernel implements hybrid architecture of the operating system.

Hybrid architecture: Advantages

1. Easy to manage due to layered approach.
2. Number of layers is not very high.
3. Kernel is small and isolated.
4. Improved security and protection.

Summary

- The components of the operating system (Kernel) are process management, memory management, I/O management, File management and Error Detection.
- OS architectures describe the functionality of various components.
- These architectures include monolithic, layered, microkernel, and hybrid architectures.
- Hybrid architecture is the most efficient architecture as it combines the advantages of all other architectures.

Thank you

Dr. K. Radhika

Professor, AI&DS Department

Mail:kradhika_aids@cbit.ac.in

Contact No.: 9346253642