Module-1

1.1 Introduction to Operating System

- An Operating System (OS) is a program that manages the computer hardware.
- It also provides a basis for Application Programs and acts as an intermediary between computer User and computer Hardware.

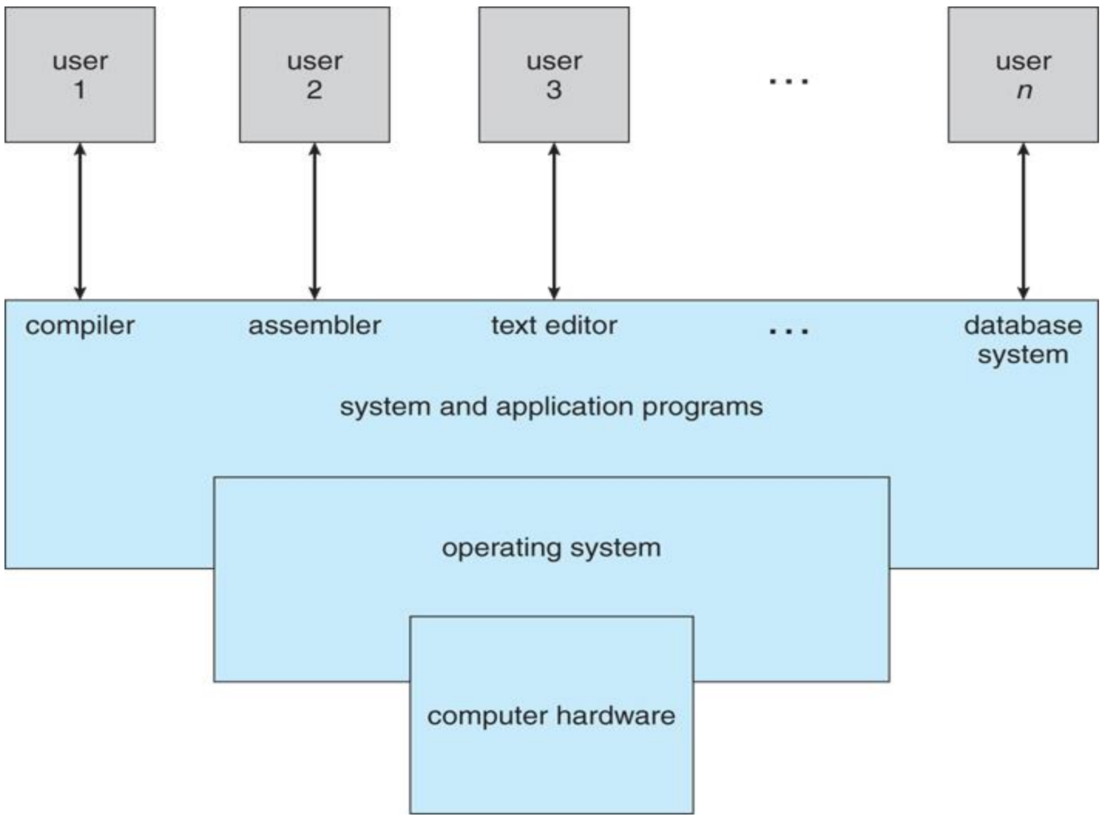


Figure.1. Abstract view of the components of a computer system

- A computer system can be divided roughly into four components:
    - Hardware
    - Operating System
    - Application programs and
    - Users

Hardware: The central processing unit (CPU), the memory (primary, secondary), and the input/output devices.

- Provides the basic computing resources for the system.

Operating System: Controls and coordinates the use of the hardware among the various application programs for the various users.

Application Programs: Such as word processor, spreadsheets, compilers, and web browsers – define the way in which these resources are used to solve user's computing problems.

1.1.1 Goals of Operating System

1) Convenience
2) Efficiency
3) Ability to Evolve

1.1.2 Functions of Operating System

- It is an interface between User and Hardware
- Allocation of resources
- Management of Memory, Security, etc.

1.1.3 Types of Operating System

- ✓ Batch OS
- ✓ Time sharing OS
- ✓ Distributed OS
- ✓ Network OS
- ✓ Real Time OS
- ✓ Multi Programming/Processing/Tasking OS

1.2 Computer System Operation

Note: Some basic knowledge of the structure of Computer System is required to understand how Operating Systems work.

- 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.

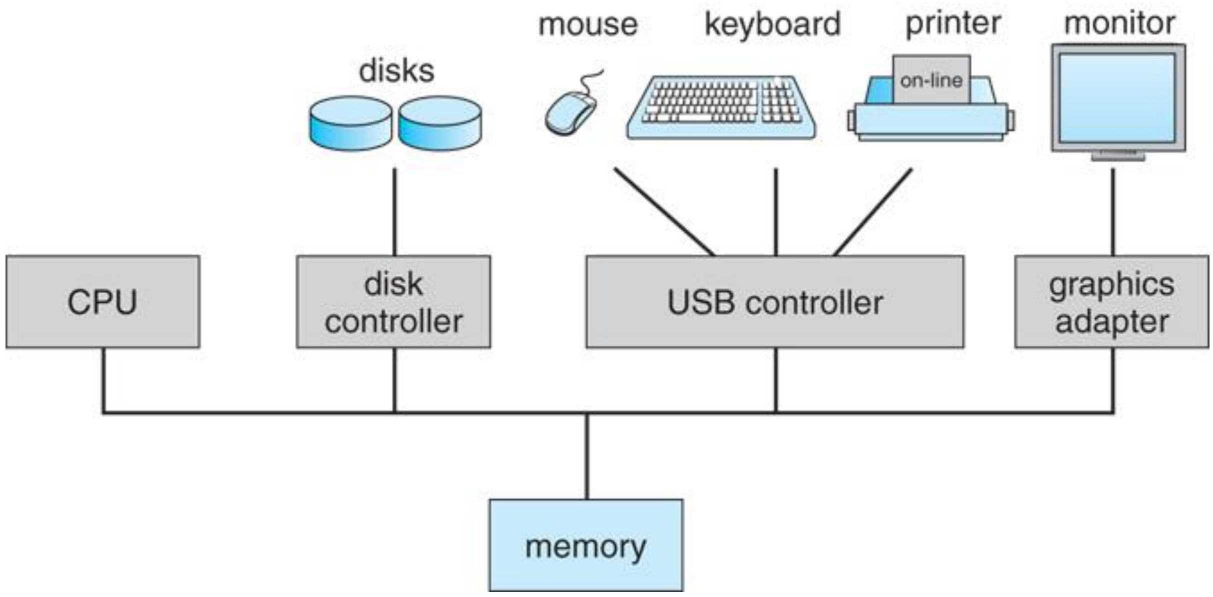


Figure.2 A Modern Computer System

- Each device controller is in charge of a specific type of device.
- The CPU and device controllers can execute concurrently, competing for memory cycles.
- To ensure orderly access to the shared memory, a memory controller is used whose function is to synchronize access to the memory.

- These entire device controllers have its own shared memory and can get access whenever it wants.

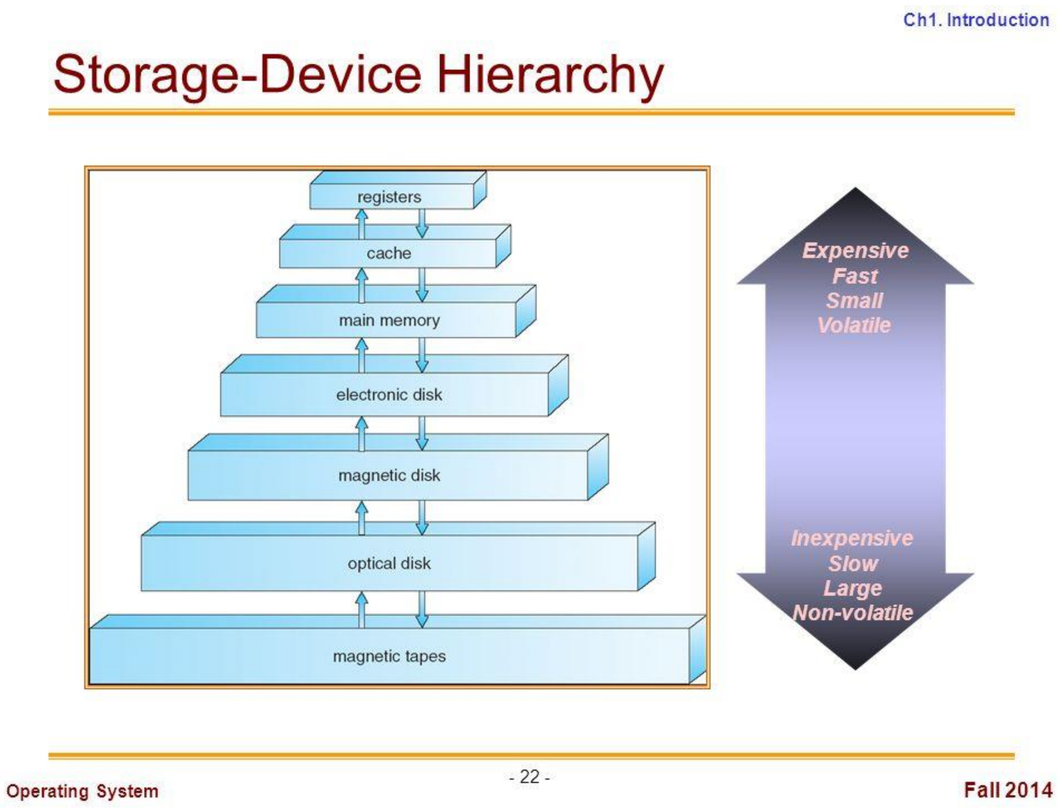
1.2.1 Some Important Terms

- **Bootstrap program**
    - The initial program that runs when a computer is powered up (or) rebooted.
    - It is stored in ROM.
    - It must know how to load the OS and start executing that system.
    - It must locate and load into memory the OS kernel.
- **Interrupt**
    - The occurrence of an event is usually signalled by an interrupt from hardware (or) software.

    - Hardware may trigger an interrupt at any time by sending a signal to the CPU, usually by the way of the system bus.
- **System call(monitor call)**
    - Software may trigger an interrupt by executing a special operation called system call.
- How an interrupt is handled?
    - When the CPU is interrupted, it stops what it is doing and immediately transfer execution to a <u>fixed location</u>.
    - The fixed location usually contains the starting address where the <u>service routine</u> of the interrupt is located.

- Then the <u>Interrupt Service Routine</u> (contains details about what the CPU as to do) executes.
- On completion, the CPU resumes the interrupted execution.

1.3 Storage Structure


Ch1. Introduction

Storage-Device Hierarchy

Operating System - 22 - Fall 2014

- Computer programs must be in main memory (RAM) to be executed.

- Secondary storage is an extension of main memory to hold large quantities of data permanently.
    - The most common secondary storage device is **magnetic disk.**

1.4 I/O Structure

- Storage is only one of many types of I/O devices within a computer.
- A large portion of OS code is dedicated to managing I/O, because of the varying nature of the devices.
- A device controller maintains some local buffer storage and a set of special purpose registers.
- The device controller is responsible for moving the data between the

peripheral devices that it controls and its local buffer storage.

- Typically, OS have a <u>device driver</u> for each device controller.
- The device driver understands the device controller and provides a uniform interface to the device to the rest of the operating system.

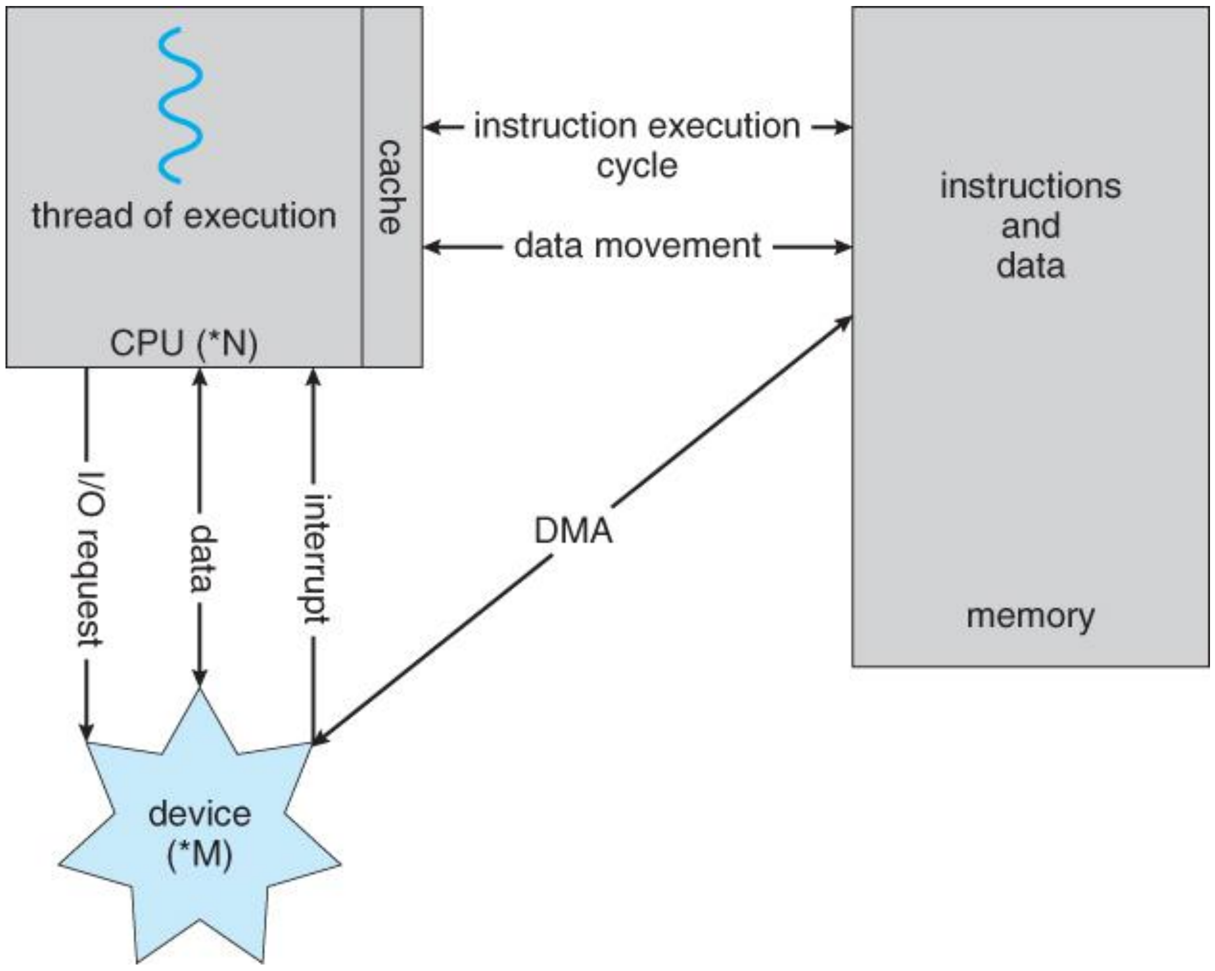


Figure.3. How modern computer system works

- To start an I/O operation, the device driver loads the appropriate registers within the device controller.
- The device controller, in turn, examines the contents of the registers to determine what action to take (such as “read a character from the keyboard”).
- The controller starts the transfer of data from the device to its local buffer.
- Once the transfer is complete, the device controller informs the device driver via an interrupt that it has finished its operation.
- The device driver then returns control to the operating system, possibly returning the data or a pointer to the data if the operation was a read.

- This form of interrupt-driven I/O is fine for moving small amounts of data but can produce high overhead when used for bulk data movements such as disk I/O.
- To solve this problem, DMA (Direct Memory Access) is used.
- After setting up buffers, pointers, and counters of the I/O device, the device controller transfers <u>entire block of data</u> directly to or form its own storage to memory, with no intervention by the CPU.
- Only one interrupt is generated per block, to tell the device driver that the operation has completed.

1.5 Computer System Architecture

Types of computer systems are based on number of general purpose processors.

- Single-Processer Systems
- Multiprocessor Systems
- Clustered Systems

Single-Processer Systems

- Most systems use single processor.
- On a single-processor system, there is one main CPU capable of executing a general purpose instruction set, including instructions from user processes.
- Almost all systems have other special-purpose processors as well.
  - They may come in the form of device-specific processors such as

keyboard, disk, and graphics controller.

Multiprocessor Systems

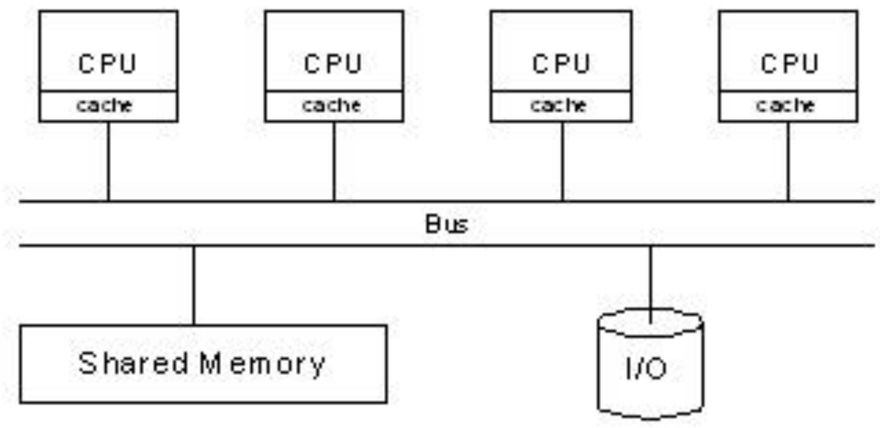
- Two or more processors in close communication, sharing the computer bus and sometimes the clock, memory, and peripheral devices.
- Also known as parallel or tightly coupled systems.
- Advantages:
    - Increased throughput – By increasing the number of processors, we expect to get more work done in less time.
    - Economy of scale – Multiprocessor systems can cost

less than multiple single processor systems, because they can share peripherals, storage, and power supplies.

  - Increased reliability – Failure of one processor will not halt the system. I,e.,
  - If we have ten processors and one fails, then each of the remaining <u>nine processors </u> can pick up a share of the work of the failed processor.

- The multiprocessor systems is of two types: symmetric and Asymmetric

Symmetric Multiprocessing (SMP)



- Each processor performs all tasks.
- All processors are peers.
- No master-slave relationship between processors.
- Example: Sun Solaris

Asymmetric Multiprocessing (SMP)

- Each processor is assigned a specific task.
- This scheme defines a master-slave relationship.

- A master processor controls the system; the other processes execute the task.
- The master processor schedules and allocated work to the slave processors.

Clustered Systems

- Like multiprocessor systems, clustered systems gather together multiple CPUs to accomplish a task.
- Used to provide high availability service.
- Clustering can be structured asymmetrically or symmetrically.
- In asymmetric clustering one machine is in hot-standby mode, while other is running the application.

- The hot-standby host machine does nothing but monitor the active server.
    - If the active server fails, then the hot-stand by host becomes the active server.

Multiprogramming & Multitasking

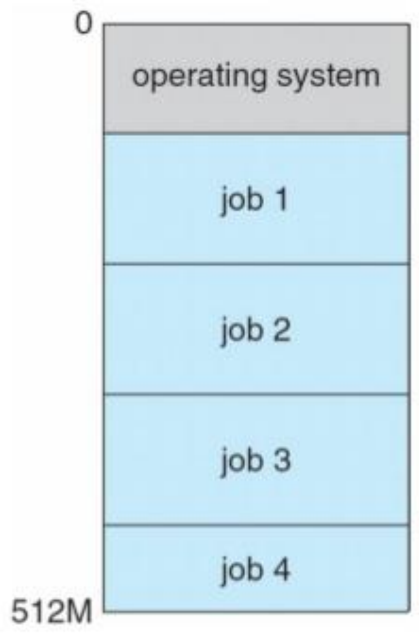
- Operating Systems vary greatly in their makeup internally.
- COMMONALITIES:
    - Multiprogramming
    - Time Sharing(Multitasking)

Multiprogramming

- Single user can't, in general, keep either the CPU (or) the I/O devices busy at all times.

- Multiprogramming increases CPU utilization by organizing jobs (code and data) so that CPU always has one to execute.

Memory Layout for Multiprogrammed System



NOTE:

Multiprogrammed systems provide an environment in which the various system resources (memory, CPU, peripheral devices) are utilized effectively, but they do not provide for user interaction with the computer system.

Timesharing (Multitasking)

- **CPU** executes multiple jobs by switching among them.
- Switches occur so frequently that the users can interact with each program while it is running.
- Time sharing requires an interactive (or hands-on)compute systems, which provides direct communication between the user and the system.
- A time-shared operating systems allows many users to share the computer simultaneously.

- Uses CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared computer.
- Each user has at least one program in memory.
- A program loaded into memory and executing is called as “<u>PROCESS</u>”.

1.6 Operating System – Services

An Operating System provides services to both the users and to the programs.

- It provides programs an environment to execute.
- It provides users the services to execute the programs in a convenient manner.

Following are a few common services provided by an operating system −

- **User Interface**
- **Program execution**
- **I/O operations**
- **File System manipulation**
- **Communication**
- **Error Detection**
- **Resource Allocation**
- **Protection**

User Interface

- Command Line Interface (CLI)
- Graphical User Interface (GUI)

Program execution

- ✓ Operating systems handle many kinds of activities from user programs to system programs.
- ✓ Each of these activities is encapsulated as a process.
- ✓ A process includes the complete execution context (code to execute, data to manipulate, registers, OS resources in use).
- ✓ Following are the major activities of an operating system with respect to program management –

1) Loads a program into memory.
2) Executes the program.
3) Handles program's execution.
4) Provides a mechanism for process synchronization.
5) Provides a mechanism for process communication.
6) Provides a mechanism for deadlock handling.

I/O Operation

- An I/O subsystem comprises of I/O devices and their corresponding driver software.
- Drivers hide the peculiarities of specific hardware devices from the users.
- An Operating System manages the communication between user and device drivers.

1) I/O operation means <u>read or write operation</u> with any <u>file or any specific I/O device</u>.
2) Operating system provides the access to the required I/O device when required.

File system manipulation

- A file represents a collection of related information.
- Computers can store files on the disk (secondary storage), for long-term storage purpose.

- Examples of storage media include magnetic tape, magnetic disk and optical disk drives like CD, DVD.
- Each of these media has its own properties like speed, capacity, data transfer rate and data access methods.
- A file system is normally organized into directories for easy navigation and usage.
- These directories may contain files and other directions.
- Following are the major activities of an operating system with respect to file management −

1) Program needs to read a file or write a file.
2) The operating system gives the permission to the program for operation on file.
3) Permission varies from read-only, read-write, denied and so on.
4) Operating System provides an interface to the user to create/delete files.
5) Operating System provides an interface to the user to create/delete directories.

6) Operating System provides an interface to create the backup of file system.

Communication

- In case of distributed systems which are a collection of processors that do not share memory, peripheral devices, or a clock, the operating system manages communications between all the processes.
- Multiple processes communicate with one another through communication lines in the network.
- The OS handles routing and connection strategies, and the problems of contention and security.
- Following are the major activities of an operating system with respect to communication −

1) Two processes often require data to be transferred between them
2) Both the processes can be on one computer or on different computers, but are connected through a computer network.

3) Communication may be implemented by two methods, either by Shared Memory or by Message Passing.

Error handling

- Errors can occur anytime and anywhere.
- An error may occur in CPU, in I/O devices or in the memory hardware.
- Following are the major activities of an operating system with respect to error handling –

1) The OS constantly checks for possible errors.
2) The OS takes an appropriate action to ensure correct and consistent computing.

Resource Management

- In case of multi-user or multi-tasking environment, resources such as main memory, CPU cycles and files storage are to be allocated to each user or job.
- Following are the major activities of an operating system with respect to resource management –

1) The OS manages all kinds of resources using schedulers.
2) CPU scheduling algorithms are used for better utilization of CPU.

Protection and Security

- Considering a computer system having multiple users and concurrent execution of multiple processes, the various processes must be protected from each other's activities.
- Protection refers to a mechanism or a way to control the access of programs, processes, or users to the resources defined by a computer system.
- Following are the major activities of an operating system with respect to protection −

1) The OS ensures that all access to system resources is controlled.
2) The OS ensures that external I/O devices are protected from invalid access attempts.
3) The OS provides authentication features for each user by means of passwords.

1.7 Functionalities of operating System

- Memory Management
- Processor Management
- Device Management
- File Management
- Security
- Control over system performance
- Job accounting
- Error detecting aids
- Coordination between other software and users

Memory Management

- Memory management refers to management of Primary Memory or Main Memory.
- Main memory is a large array of words or bytes where each word or byte has its own address.
- Main memory provides a fast storage that can be accessed directly by the CPU.

- For a program to be executed, it must in the main memory.
- An Operating System does the following activities for memory management −

1) Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.
2) In multiprogramming, the OS decides which process will get memory when and how much.
3) Allocates the memory when a process requests it to do so.
4) De-allocates the memory when a process no longer needs it or has been terminated.

Processor Management

- In multiprogramming environment, the OS decides which process gets the processor when and for how much time.
- This function is called **process scheduling**.
- An Operating System does the following activities for processor management −

1) Keeps tracks of processor and status of process. The program responsible for this task is known as **traffic controller**.
2) Allocates the processor (CPU) to a process.
3) De-allocates processor when a process is no longer required.

Device Management

- An Operating System manages device communication via their respective drivers.
- It does the following activities for device management −

1) Keeps tracks of all devices. Program responsible for this task is known as the **I/O controller**.
2) Decides which process gets the device when and for how much time.
3) Allocates the device in the efficient way.
4) De-allocates devices.

File Management

- A file system is normally organized into directories for easy navigation and usage.

- These directories may contain files and other directions.
- An Operating System does the following activities for file management –

1) Keeps track of information, location, uses, status etc. The collective facilities are often known as **file system**.
2) Decides who gets the resources.
3) Allocates the resources.
4) De-allocates the resources.

Other Important Activities

Following are some of the important activities that an Operating System performs –

- **Security** – By means of password and similar other techniques, it prevents unauthorized access to programs and data.
- **Control over system performance** – Recording delays between request for a service and response from the system.
- **Job accounting** – Keeping track of time and resources used by various jobs and users.

- **Error detecting aids** – Production of dumps, traces, error messages, and other debugging and error detecting aids.
- **Coordination between other software's and users** – Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

1.8 Types of Operating System

Operating systems are there from the very first computer generation and they keep evolving with time.

Batch operating system

- The users of a batch operating system do not interact with the computer directly.
- Each user prepares his job on an off-line device like punch cards and submits it to the computer operator.
- To speed up processing, jobs with similar needs are batched together and run as a group.

- The programmers leave their programs with the operator and the operator then sorts the programs with similar requirements into batches.

The problems with Batch Systems are as follows −

- Lack of interaction between the user and the job.
- CPU is often idle, because the speed of the mechanical I/O devices is slower than the CPU.
- Difficult to provide the desired priority.

Time-sharing operating systems

- Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time.
- Time-sharing or multitasking is a logical extension of multiprogramming.
- Processor's time which is shared among multiple users simultaneously is termed as time-sharing.
- The main difference between Multiprogrammed Batch Systems and Time-

Sharing Systems is that in case of Multiprogrammed batch systems, the objective is to maximize processor use, whereas in Time-Sharing Systems, the objective is to minimize response time.

- Multiple jobs are executed by the CPU by switching between them, but the switches occur so frequently.
- Thus, the user can receive an immediate response.
- For example, in a transaction processing, the processor executes each user program in a short burst or quantum of computation. That is, if **n** users are present, then each user can get a time quantum. When the user submits the command, the response time is in few seconds at most.
- The operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time.
- Computer systems that were designed primarily as batch systems have been modified to time-sharing systems.

Advantages

- Provides the advantage of quick response.
- Avoids duplication of software.
- Reduces CPU idle time.

Disadvantages

- Problem of reliability.
- Question of security and integrity of user programs and data.
- Problem of data communication.

Distributed operating System

- Distributed systems use multiple central processors to serve multiple real-time applications and multiple users.
- Data processing jobs are distributed among the processors accordingly.
- The processors communicate with one another through various communication lines (such as high-speed buses or telephone lines).
- These are referred as **loosely coupled systems** or distributed systems.
- Processors in a distributed system may vary in size and function.

- These processors are referred as sites, nodes, computers, and so on.

Advantages

- With resource sharing facility, a user at one site may be able to use the resources available at another.
- Speedup the exchange of data with one another via electronic mail.
- If one site fails in a distributed system, the remaining sites can potentially continue operating.
- Better service to the customers.
- Reduction of the load on the host computer.
- Reduction of delays in data processing.

Network operating System

- A Network Operating System runs on a server and provides the server the capability to manage data, users, groups, security, applications, and other networking functions.
- The primary purpose of the network operating system is to allow shared file and printer access among multiple computers in a

network, typically a local area network (LAN), a private network or to other networks.

Examples:

Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD.

Advantages

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

Disadvantages

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

Real Time operating System

- A real-time system is defined as a data processing system in which the time interval required to process and respond to inputs is so small that it controls the environment.

- The time taken by the system to respond to an input and display of required updated information is termed as the **response time**. So in this method, the response time is very less as compared to online processing.
- Real-time systems are used when there are rigid time requirements on the operation of a processor or the flow of data and real-time systems can be used as a control device in a dedicated application.
- A real-time operating system must have well-defined, fixed time constraints, otherwise the system will fail.

  Example: Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

There are two types of real-time operating systems.

Hard real-time systems

- Hard real-time systems guarantee that critical tasks complete on time.

- In hard real-time systems, secondary storage is limited or missing and the data is stored in ROM.
- In these systems, virtual memory is almost never found.

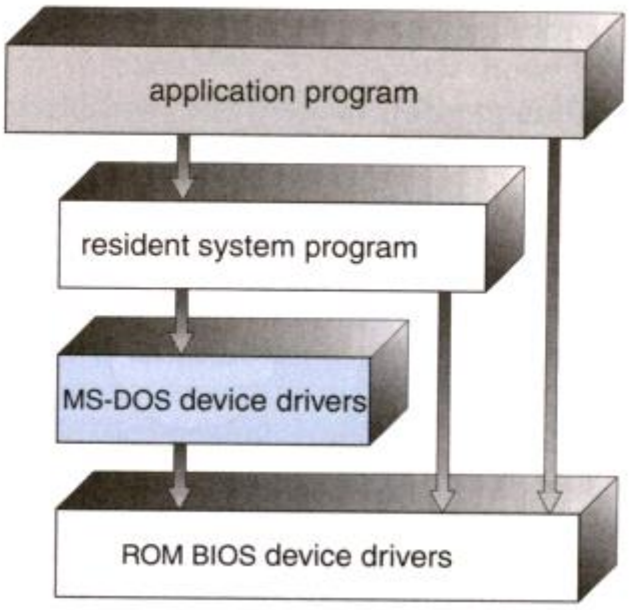
Soft real-time systems

- Soft real-time systems are less restrictive.
- A critical real-time task gets priority over other tasks and retains the priority until it completes.
- Soft real-time systems have limited utility than hard real-time systems.

  Example: Multimedia, Virtual Reality, Advanced Scientific Projects like undersea exploration and planetary rovers, etc.

1.9 Structure of Operating System

1) Simple Structure

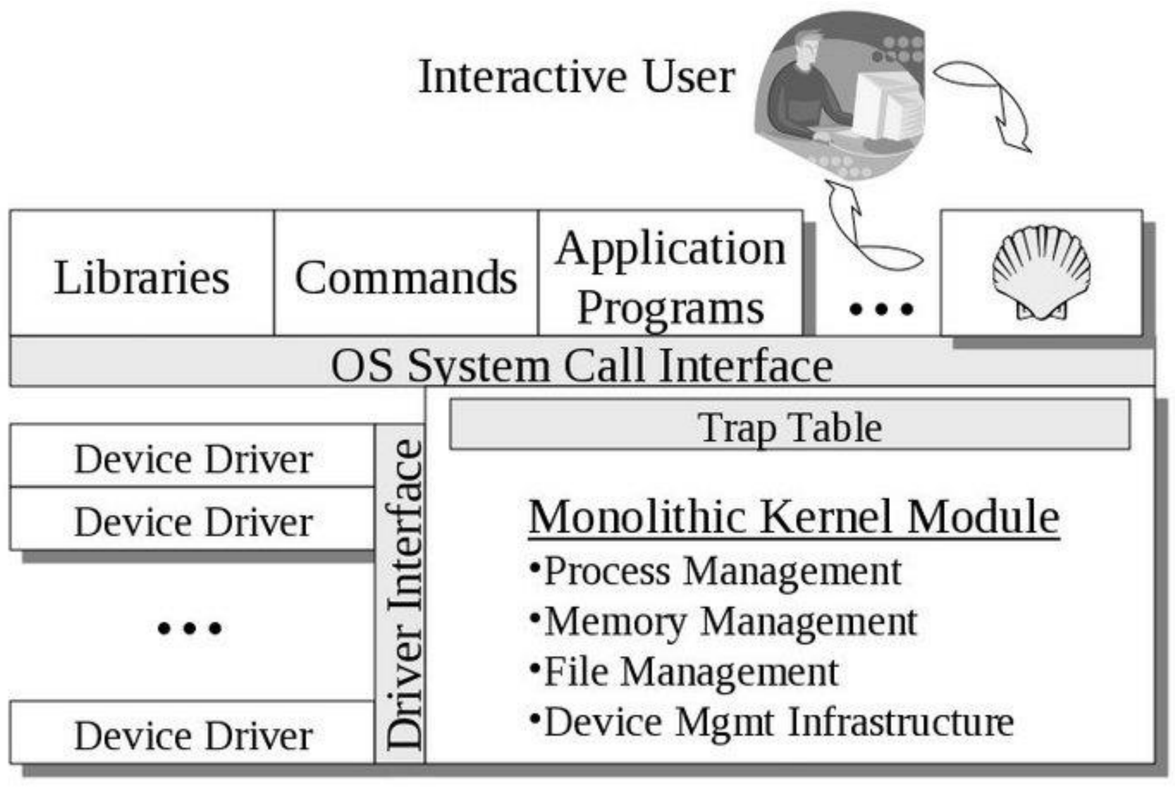


In MS-DOS, applications may bypass the operating system.

- Operating systems such as MS-DOS and the original UNIX did not have well-defined structures.
- There was no CPU Execution Mode (user and kernel), and so errors in applications could cause the whole system to crash.

2) Monolithic Approach

- Functionality of the OS is invoked with simple function calls within the kernel, which is one large program.
- Device drivers are loaded into the running kernel and become part of the kernel.



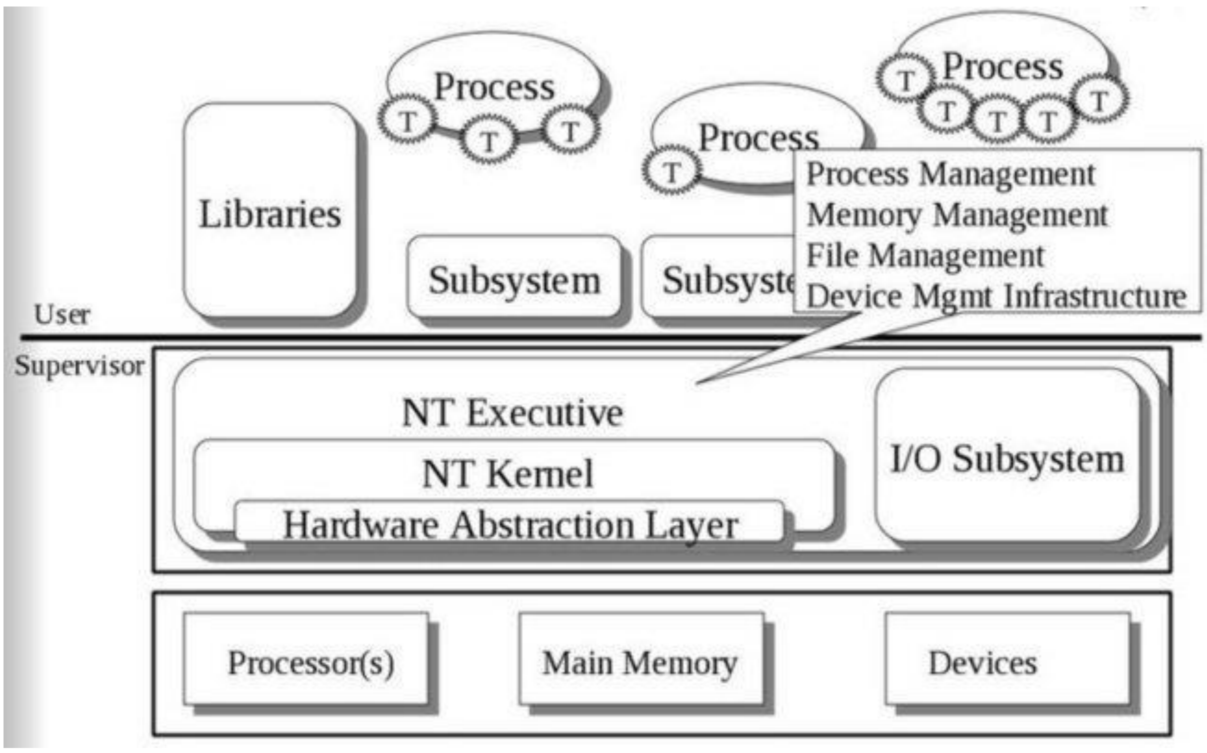
A monolithic kernel, such as Linux and other Unix systems.

3) Layered Approach

This approach breaks up the operating system into different layers.

- This allows implementers to change the inner workings, and increases modularity.
- As long as the external interface of the routines don't change, developers have more freedom to change the inner workings of the routines.

- With the layered approach, the bottom layer is the hardware, while the highest layer is the user interface.
    - The main *advantage* is simplicity of construction and debugging.
    - The main *difficulty* is defining the various layers.
    - The main *disadvantage* is that the OS tends to be less efficient than other implementations.



The Microsoft Windows NT Operating System. The lowest level is a monolithic kernel, but many OS components are at a higher level, but still part of the OS.

4) Microkernels

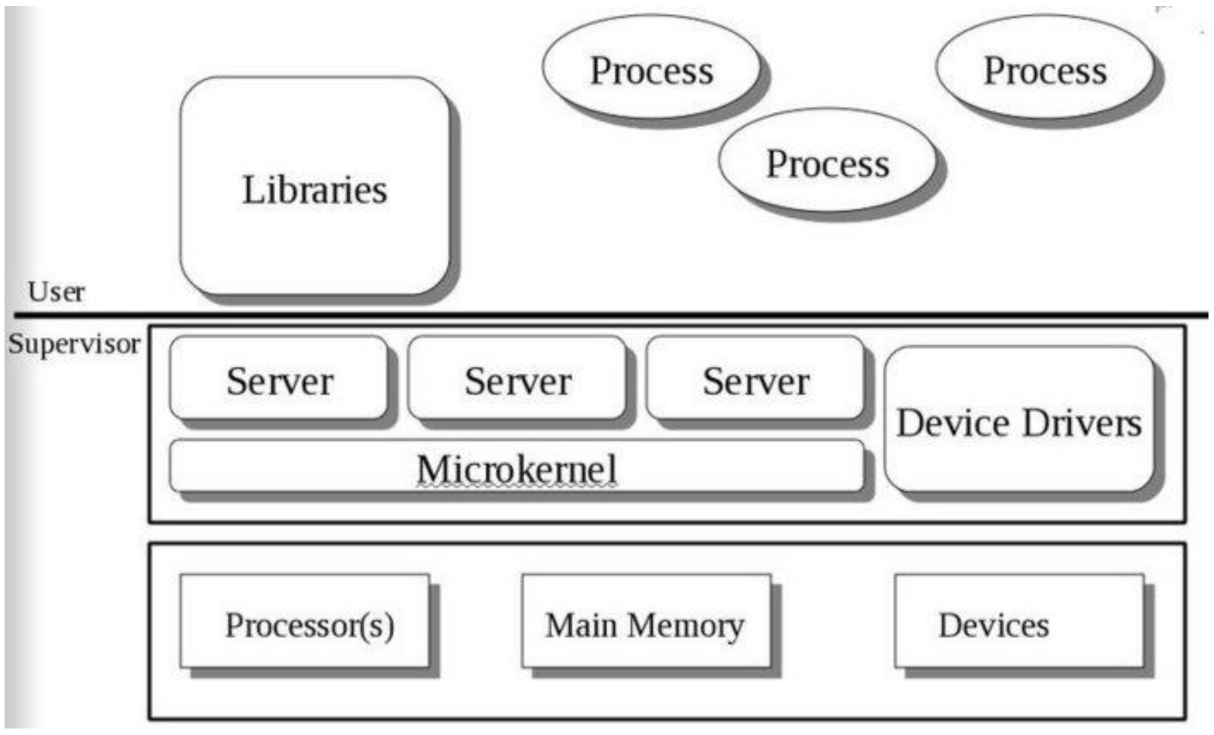
This structures the operating system by removing all nonessential portions of the kernel and implementing them as system and user level programs.

- Generally they provide minimal process and memory management, and a communications facility.
- Communication between components of the OS is provided by message passing.

The *benefits* of the microkernel are as follows:

- Extending the operating system becomes much easier.
- Any changes to the kernel tend to be fewer, since the kernel is smaller.
- The microkernel also provides more security and reliability.

Main *disadvantage* is poor performance due to increased system overhead from message passing.


Process
Process
Process
Libraries
User
Supervisor
Server
Server
Server
Device Drivers
Microkernel
Processor(s)
Main Memory
Devices