UNIT I

What is an Operating System?

A program that acts as an intermediary/interface between a user of a computer and the computer hardware.
OR An operating System is a collection of system programs that together control the operations of a computer system.

Some examples of operating systems are UNIX, Mach, MS-DOS, MS-Windows, Windows/NT, Chicago, OS/2,
MacOS, VMS, MVS, and VM.

Operating system performs the following functions:

1. Booting

Booting is a process of starting the computer operating system starts the computer to work. It checks the computer and makes it ready to work.

2. Memory Management
It is also an important function of operating system. The memory cannot be managed without operating system. Different programs and data execute in memory at one time. if there is no operating system, the programs may mix with each other. The system will not work properly.

3. Loading and Execution
A program is loaded in the memory before it can be executed. Operating system provides the facility to load programs in memory easily and then execute it.

4. Data security
Data is an important part of computer system. The operating system protects the data stored on the computer from illegal use, modification or deletion.

5. Disk Management
Operating system manages the disk space. It manages the stored files and folders in a proper way.
6. Process Management

CPU can perform one task at one time. if there are many tasks, operating system decides which task should get the CPU.

7. Device Controlling
operating system also controls all devices attached to computer. The hardware devices are controlled with the help of small software called device drivers..

8. Providing interface

It is used in order that user interface acts with a computer mutually. User interface controls how you input data and instruction and how information is displayed on screen. The operating system offers two types of the interface to the user:
1. Graphical-line interface: It interacts with of visual environment to communicate with the computer. It uses windows, icons, menus and other graphical objects to issues commands.

2. Command-line interface: it provides an interface to communicate with the computer by typing Commands.

Components of an Operating System

1. Kernel

- Core part of the OS
- Manages CPU, memory, devices, and processes
- Always in memory

2. Shell

- Interface between users and kernel
- Can be command-line or graphical

3. File System

- Defines how files are named, stored, retrieved, and organized

4. System Programs

- Utilities like compilers, editors, file managers, etc.

OPERATING SYSTEM SERVICES

Operating systems provide a set of **services** to make the system **easy to use**, **efficient**, and **reliable**. These services help both **users** and **programs (processes)**.

---

1. Program Execution

The OS loads a program into memory and executes it.

Functions:

- Load program into RAM
- Start program execution
- Handle errors during execution
- Terminate the program safely

---

2. I/O Operations

Programs need I/O to read/write data.

OS Services:

- Handle device communication
- Provide device drivers
- Manage keyboard, mouse, printer, disks, etc.

Example: Reading a file from disk.

---

3. File System Manipulation

The OS manages files and directories.

Functions:

- Create, delete files and folders
- Read, write, modify files
- Search and list files
- Manage permissions (read/write/execute)

---

4. Communication (Interprocess Communication — IPC)

Processes may need to exchange data.

Methods:

- Shared memory
- Message passing
- Signals
- Pipes and sockets

Used in:

- Client-server systems
- Distributed systems

---

5. Error Detection

OS constantly monitors for errors.

Errors include:

- Memory errors
- I/O device failures
- CPU faults
- Program errors (divide by zero)

OS Role:

- Detect
- Report
- Take corrective actions

---

6. Resource Allocation

OS acts as a **resource manager**.

Resources include:

- CPU time
- Memory space
- Disk space
- I/O devices

Uses scheduling algorithms like:

- Round Robin
- FCFS
- Priority Scheduling

---

7. Accounting

OS keeps track of resource usage.

Tracks:

- CPU usage
- Disk usage
- Memory usage
- I/O operations

Used for:

- Billing in cloud systems
- Performance monitoring

---

8. Protection and Security

Protects data and system integrity.

Protection

Controls access between processes.

Security

Protects against:

- Unauthorized access
- Malware
- Data theft
- Network attacks

Mechanisms:

- Passwords
- Encryption
- User authentication

⭐ ADDITIONAL OS SERVICES (MODERN SYSTEMS)

9. User Interface (UI)

Two types:

- **Command Line Interface (CLI)**
- **Graphical User Interface (GUI)** (Windows, Linux Desktop, macOS)

---

10. Networking Services

- Internet access
- Protocol support (TCP/IP)
- Network resource sharing

---

11. System Monitoring & Performance

- Task manager
- Activity monitor
- System logs

OBJECTIVES OF O.S (GOALS)

The OS has 3 main objectives.

- **Convenience.** An OS makes a computer more convenient to the user for using. (Easy-to-use commands, graphical user interface(GUI))
- **Efficiency.** An OS allows the computer system resources to be used in an efficient manner, to ensure good resource utilization efficiency, and provide appropriate corrective actions when it becomes low.
- **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 without interfering with service.

Computer System Architecture

Computer system can be divided into four components

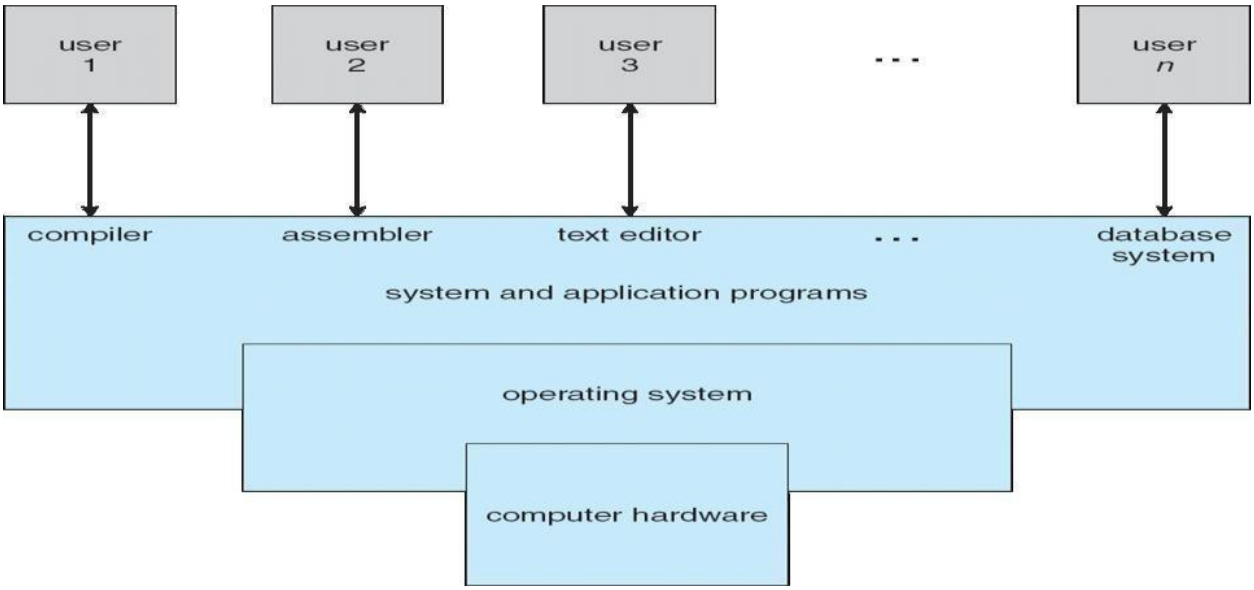
1. Hardware – provides basic computing resources
   CPU, memory, I/O devices.

2 Operating system -Controls and coordinates use of hardware among various applications and users

3 Application programs – define the ways in which the system resources are used to solve the computing problems of the users
Word processors, compilers, web browsers, database systems, video games

4. Users -People

See the diagram below of Computer System Architecture



Computer architecture means construction/design of a computer. A computer system may be organized in different ways. Some computer systems have single processor and others have multiprocessors. So based on the processors used in computer systems, they are categorized into the following systems.

1. Single-processor system

2. Multiprocessor system

3. Clustered Systems:

1. Single-Processor Systems:

Some computers use only one processor such as microcomputers (or personal computers PCs). On a single-processor system, there is only one CPU that performs all the activities in the computer system. However, most of these systems have other special purpose processors, such as I/O processors that move data quickly among different components of the computers. These processors execute only a limited system programs and do not run the user program.

Sometimes they are managed by the operating system. Similarly, PCs contain a special purpose microprocessor in the keyboard, which converts the keystrokes into computer codes to be sent to the CPU. The use of special purpose microprocessors is common in microcomputer. But it does not mean that this system is multiprocessor. A system that has only one general-purpose CPU, is considered as single- processor system.

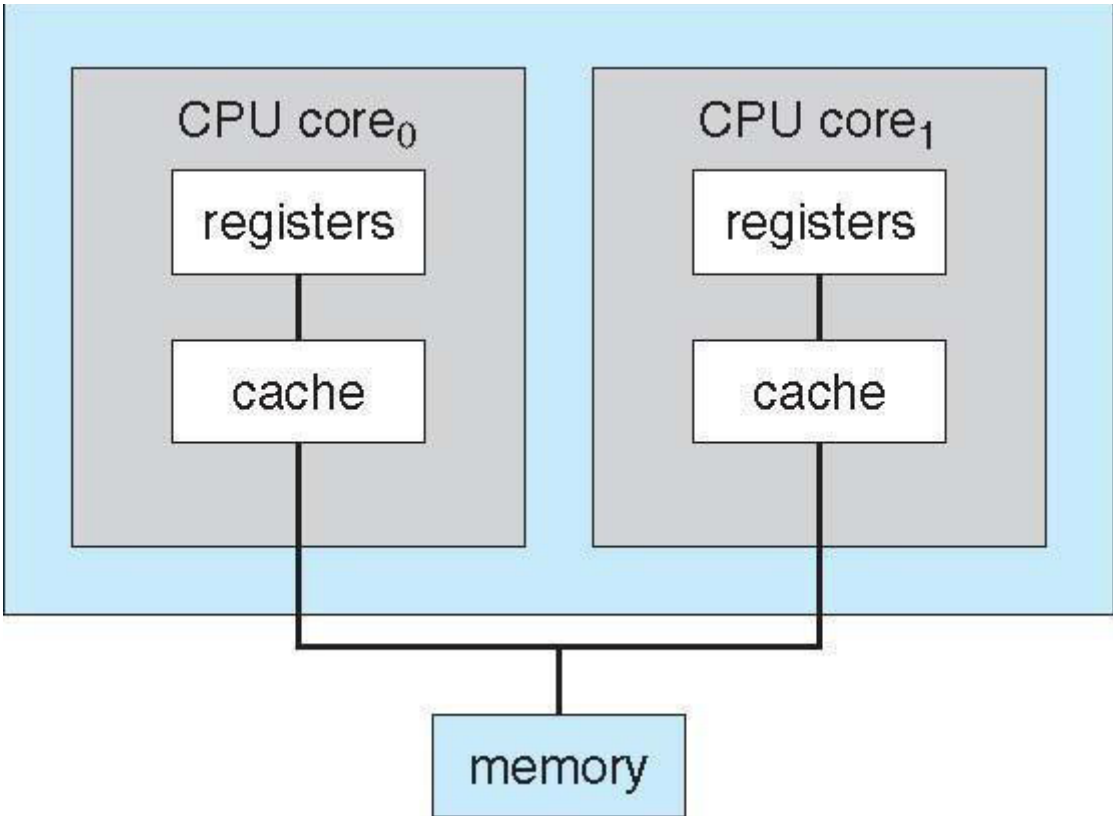
2. Multiprocessor Systems:

In multiprocessor system, two or more processors work together. In this system, multiple programs (more than one program) are executed on different processors at the same time. This type of processing is known as multiprocessing. Some operating systems have features of multiprocessing. UNIX is an example of multiprocessing operating system. Some versions of Microsoft Windows also support multiprocessing.
Multiprocessor system is also known as parallel system. Mostly the processors of multiprocessor system share the common system bus, clock, memory and peripheral devices. This system is very fast in data processing.
Types of Multiprocessor Systems:
The multiprocessor systems are further divided into two types; (i). Asymmetric multiprocessing system (ii). Symmetric multiprocessing system
(i) Asymmetric Multiprocessing System (AMS):

The multiprocessing system, in which each processor is assigned a specific task, is known as Asymmetric Multiprocessing System. For example, one processor is dedicated for handling user's requests, one processor is dedicated for running application program, and one processor is dedicated for running image processing and so on. In this system, one processor works as master processor, while other processors work as slave processors. The master processor controls the operations of system. It also schedules and distributes tasks among the slave processors. The slave processors perform the predefined tasks.
(ii) Symmetric Multiprocessing System (SMP):

The multiprocessing system, in which multiple processors work together on the same task, is known as Symmetric Multiprocessing System. In this system, each processor can perform all types of tasks. All processors are treated equally and no master-slave relationship exists between the processors

For example, different processors in the system can communicate with each other. Similarly, an I/O can be processed on any processor. However, I/O must be controlled to ensure that the data reaches the appropriate processor. Because all the processors share the same memory, so the input data given to the processors and their results must be separately controlled. Today all modern operating systems including Windows and Linux provide support for SMP.
It must be noted that in the same computer system, the asymmetric multiprocessing and symmetric multiprocessing technique can be used through different operating systems.



A Dual-Core Design

3. Clustered Systems:
Clustered system is another form of multiprocessor system. This system also contains multiple processors but it differs from multiprocessor system. The clustered system consists of two or more individual systems that are coupled together. In clustered system, individual systems (or clustered computers) share the same storage and are linked together ,via Local Area Network (LAN).
A layer of cluster software runs on the cluster nodes. Each node can monitor one or more of the other nodes over the LAN. If the monitored machine fails due to some technical fault (or due to other reason), the monitoring machine can take ownership of its storage. The monitoring machine can also restart the applications that were running on the failed machine. The users of the applications see only an interruption of service.

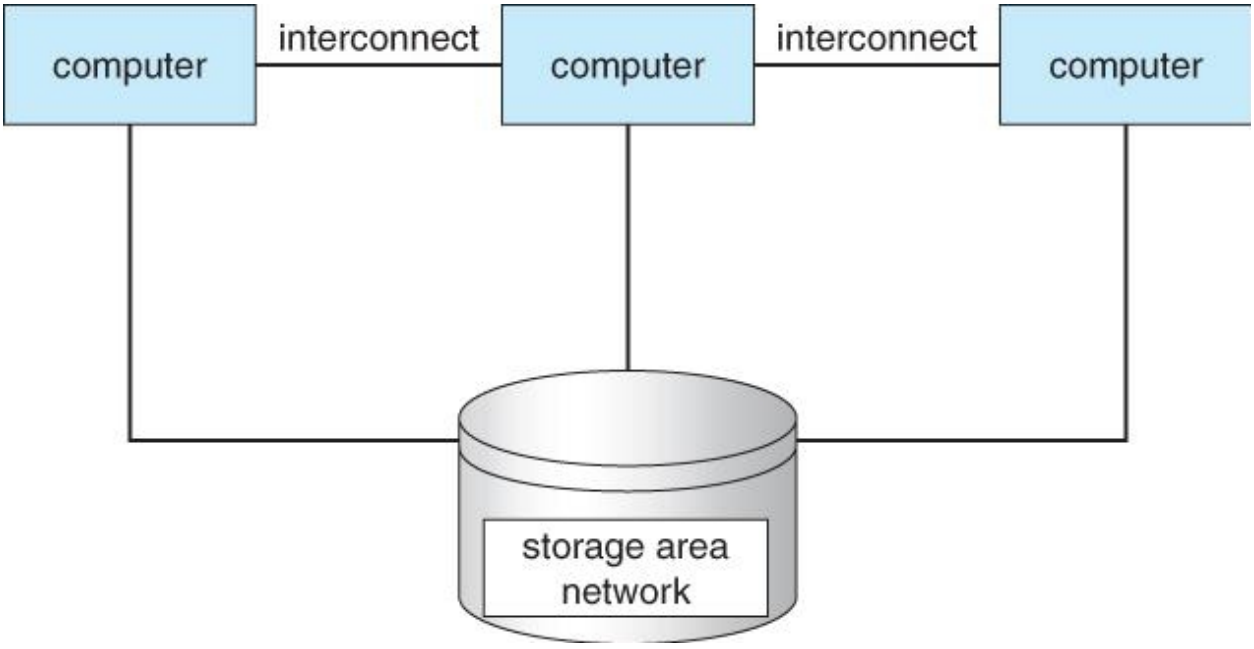
Types of Clustered Systems:

Like multiprocessor systems, clustered system can also be of two types (i). Asymmetric Clustered System
(ii). Symmetric Clustered System
(i). Asymmetric Clustered System:

In asymmetric clustered system, one machine is in hot-standby mode while the other

machine is running the application. The hot-standby host machine does nothing. It only monitors the active server. If the server fails, the hot-standby machine becomes the active server.
(ii). Symmetric Clustered System:

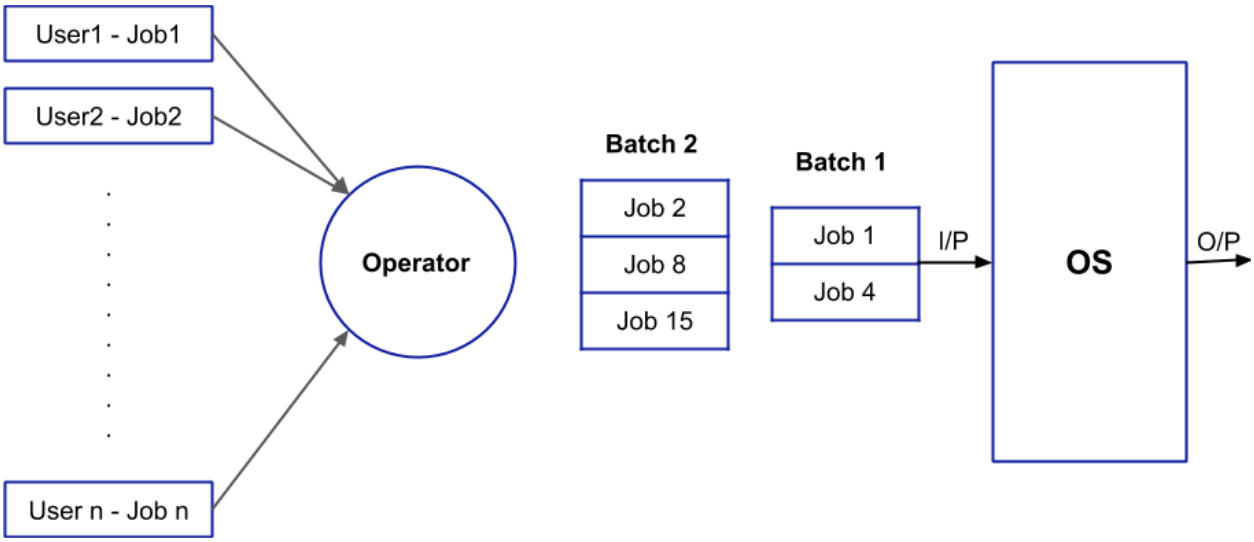
In symmetric clustered system, multiple hosts (machines) run the applications. They also monitor each other. This mode is more efficient than asymmetric system, because it uses all the available hardware. This mode is used only if more than one application be available to run.



TYPES OF OPERATING SYSTEM.

Batch Operating System

A Batch Operating System is designed to handle large groups of similar jobs efficiently. It does not interact with the computer directly but instead processes jobs that are grouped by an operator. These jobs are queued and executed one after the other, without user interaction during the process.



Advantages of Batch Operating System

- **Efficient Job Management**: Multiple users can efficiently share the system, making it cost-effective.
- **Minimal Idle Time**: The system minimizes idle time by processing jobs in a continuous sequence without human intervention.
- **Handling Repetitive Tasks**: Ideal for managing large, repetitive tasks, such as payroll and billing, with minimal effort.
- **Improved Throughput**: Batch systems can handle high volumes of jobs at once, improving overall system throughput.

Disadvantages of Batch Operating System

- **Inefficient CPU Utilization**: When a job is waiting for input/output (I/O), the CPU remains idle, leading to poor utilization of resources.
- **Unpredictable Job Completion**: If one job fails, others may be delayed indefinitely, making job completion time unpredictable.
- **Increased Response Time**: The time between job submission and output can be high as all jobs are processed sequentially.
- **Lack of Real-Time Feedback**: Users cannot interact with the system in real-time, making it less suitable for interactive tasks.

Examples:

Payroll Systems

Bank Statements

2. Multi-Programming Operating System

In a Multi-Programming Operating System, multiple programs run in memory at the same time. The CPU switches between programs, utilizing its resources more effectively and improving overall system performance.

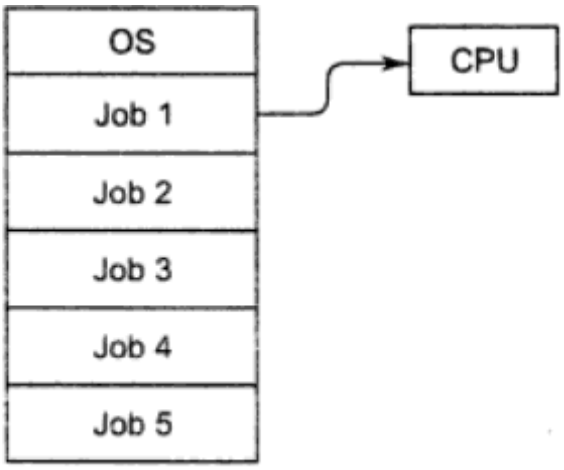


Fig. *Multiprogramming*

In non-multiprogramming system, the CPU can execute only one program at a time, if the running program waiting for any I/O device, the CPU becomes idle, so it will effect on the performance of the CPU.

But in multiprogramming environment, any I/O wait happened in a process, then the CPU switches from that job to another job in the job pool. If enough jobs could be held in main memory at once, the CPU is not idle at any time.

For Example: The idea is common in other life situations. The doctor does not have only one patient at a time, number of patients reside in the hospital under treatment. If the doctor has enough patients a doctor never needs to be idle.

Advantages of Multi-Programming Operating System

- CPU is better utilized and the overall performance of the system improves.
- It helps in reducing the response time.

Distributed Systems

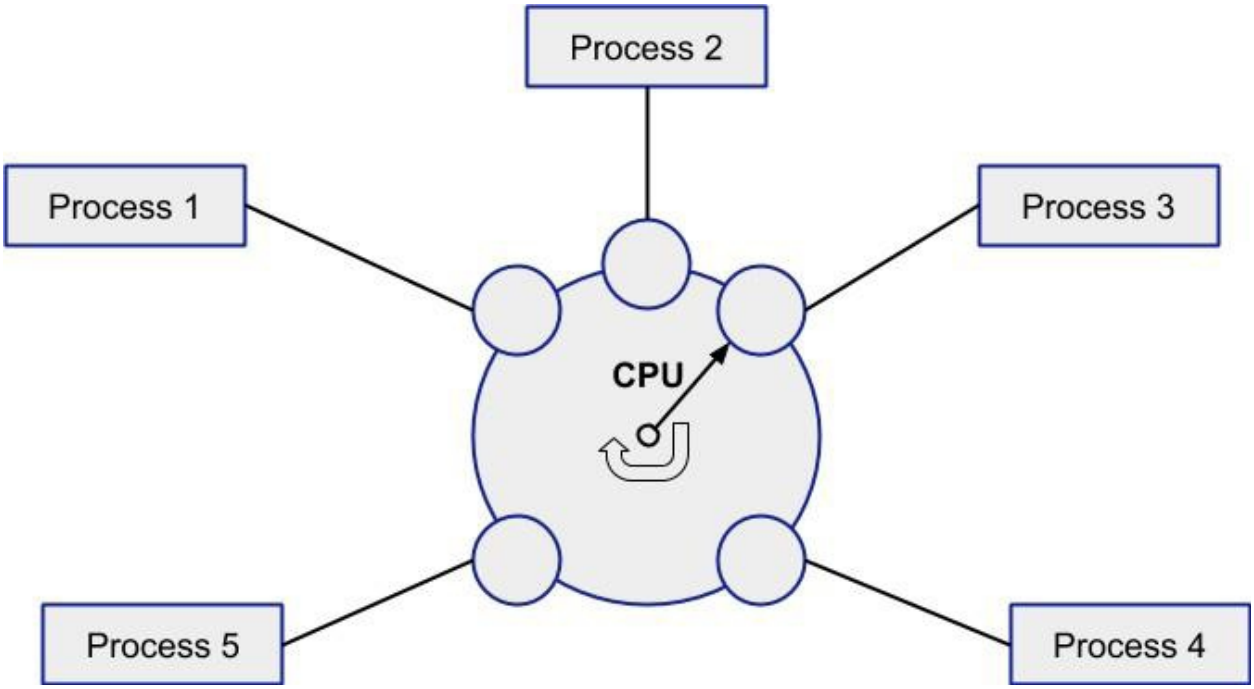
A recent trend in computer system is to distribute computation among several processors. The processors in distribute system may vary in size and function, and referred by a number of different names such as sites, nodes, computers and so on depending on the context.

A distributed system is basically a collection of autonomous (independent by function) computer systems which co-operate with one another through their hardware and software interconnections.

In distributed systems, the processors cannot share memory or time, each processor has its own local memory. The processors communicate with one another through various communication lines such as high speed buses .These systems are also called as *"Loosely Coupled systems"*.

Time sharing or multitasking is a logical extension of multiprogramming. In time sharing environment, a number of jobs are loaded on to the memory and a number of users are communicating with the computer through different terminals. The OS allocates a fixed time interval (TIME SLICE) to each program in memory. Thus each program in memory is executed for a fixed interval of time.

As soon as the time allotted for a particular program is completed, the CPU starts executing the next program. This process is continued till all the programs in the memory are executed. A program may need number of time slices for its complete execution. Although the computer system is executing one job at a time, due to the speed of the CPU, every user on a terminal has the feeling that his program that is being executed continuously, because, after every time slice, the user gets a response from the computer. The user on the terminal is communicating with his running program, and is able to debug and experiment with his program.



In time sharing os,CPU has efficient utilazation.

Parallel Systems

Almost all the systems are uni-processor systems *i.e.*, they have only one CPU. Systems in which there are more than one CPU is called as Multi-processor systems. These systems have been developed to enhance the computing power of a computing system, and the features of this system is that, they share the memory, bus and the peripheral devices. These systems are referred as *"Tightly coupled systems"*. A system consisting of more than one processor and it is a tightly coupled, then the system is called "Parallel system".

In parallel systems number of processors executing their jobs in parallel (simultaneous process). Multi-processor systems are divided into following categories:

- Symmetric
- Asymmetric

In symmetric multi-processing, each processor runs a shared copy of operating system . The processors can communicate with each other and execute these copies concurrently. Thus, in a symmetric system, all the processors share an equal amount of load .Encore's version of UNIX for the Multimax computer is an example of symmetric multiprocessing. In this system various processors execute copies of UNIX operating system, thereby executing M processes if there are M processors.

Asymmetric multi-processing is based on the principle of master-slave relationship. In this system, one of the processors runs the operating system and that processor is called the master processor. Other processors run user processes and are known as slave processors. In other words, the master processor controls, schedules and allocates the task to the slave processors. Asymmetric multi-processing is more common in extremely large systems, where one of the time consuming tasks is processing I/O requests. In the asymmetric systems the processors do not share the equal load.

Advantages:

1. It results in saving money compared to the stand alone systems, since CPU'S can share memory, bus and peripherals.
2. Throughput can be increased
3. They increase the reliability.

Since there are more than one CPU, the failure of one or more of the CPU does not halt the entire system, but only slows down the work. For example, if there are five processors, all the five working together gives full efficiency. If two CPU's fail, then the system still works but only at 60% efficiency. This indicates increased aspect of reliability compared to stand alone systems.

REAL-TIME OPERATING SYSTEM (RTOS)

A **Real-Time Operating System** is an operating system designed to **process data and respond to inputs instantly**, within a **guaranteed time frame**.

RTOS focuses on **timeliness**, **predictability**, and **deterministic behavior**.

TYPES OF REAL-TIME OPERATING SYSTEMS

Hard Real-Time Operating System.

A **Hard Real-Time OS** is an operating system where **tasks must be completed within a strict and guaranteed time limit**.
Missing a deadline can lead to **catastrophic failure**, including loss of life or system damage.

Characteristics

- Deadlines **must NEVER be missed**
- Highly predictable timing
- Used in safety-critical systems

Examples of Hard Real-Time Systems

- Pacemakers
- Airbag control systems
- Industrial robots
- Missile guidance systems

Soft Real-Time Operating System

A **Soft Real-Time OS** is an operating system where **meeting deadlines is important but not absolutelymandatory**.
Missing a deadline **does not cause system failure**, but may reduce performance.

Characteristics

- Deadlines are **preferred**, not strict
- System can tolerate small delays
- Less expensive and easier to design

Examples of Soft Real-Time Systems

- Video streaming
- Online games
- Multimedia systems
- Desktop applications requiring fast response

System Calls

A **system call** is a mechanism that allows a user-level program to request a service from the operating system (OS). System calls act as a bridge between user-level applications and the low-level functions of the OS, such as file management, process control, and network communication. When a program needs to perform an operation that it cannot execute directly, it uses a system call to request the OS to perform it on its behalf.

How System Calls Work

1. **Request**: A user application makes a request for a system service (e.g., reading a file).
2. **Trap to Kernel**: The request triggers a trap or interrupt that transfers control to the kernel (OS core).
3. **Execution in Kernel Mode**: The kernel performs the requested operation with higher privileges, ensuring security and access control.
4. **Return to User Mode**: Once completed, control returns to the user application with any output from the system call.

ASSIGNMENT 01.

Why use APIs rather than system calls?

Types of System Calls

1. Process control
2. File management
3. Device management
4. Information maintenance
5. Communications

Process control

A running needs to halt its execution either normally or abnormally.
If a system call is made to terminate the running program, a dump of memory is sometimes taken and an error message generated which can be diagnosed by a debugger
`end, abort`

load, execute
create process, terminate process
get process attributes, set process attributes
wait for time
wait event, signal event
allocate and free memory

File management
OS provides an API to make these system calls for managing files
create file, delete file
open, close file
read, write, reposition
get and set file attributes

Device management

Process requires several resources to execute, if these resources are available, they will be granted and control retuned to user process. Some are physical such as video card and other such as file. User program request the device and release when finished
request device, release device
read, write, reposition
get device attributes, set device attributes
logically attach or detach devices

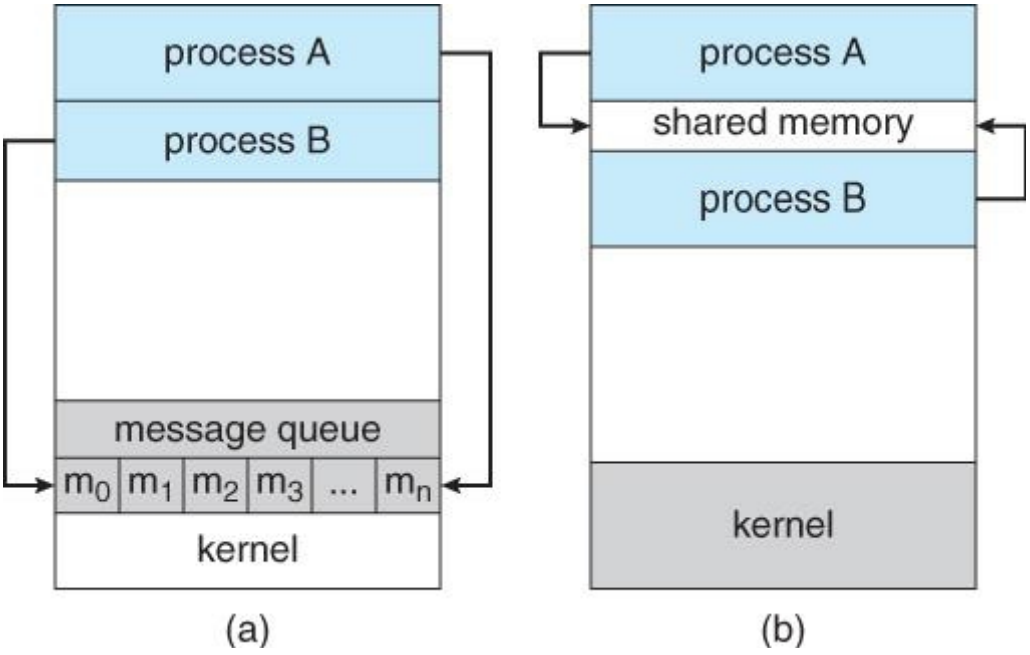
Information maintenance

System calls exist purely for transferring information between the user program and OS. It can return information about the system, such as the number of current users, the version number of the operating system, the amount of free memory or disk space and so on.
`get time or date, set time or date`
`get system data, set system data`
get and set process, file, or device attributes

INTER-COMMUNICATIONS

Two common models of communication
Message-passing model, information is exchanged through an inter process-communication facility provided by the OS.
Shared-memory model, processes use map memory system calls to gain access to regions of memory owned by other processes.
create, delete communication connection

send, receive messages
transfer status information
attach and detach remote devices



Process
A process is a program at the time of execution.

Differences between Process and process	**Program program**
Process is a dynamic object	Program is a static object
Process is sequence of instruction execution	Program is a sequence of instructions
Process loaded in to main memory	Program loaded into secondary storage devices
Time span of process is limited	Time span of program is unlimited
Process is a active entity	Program is a passive entity

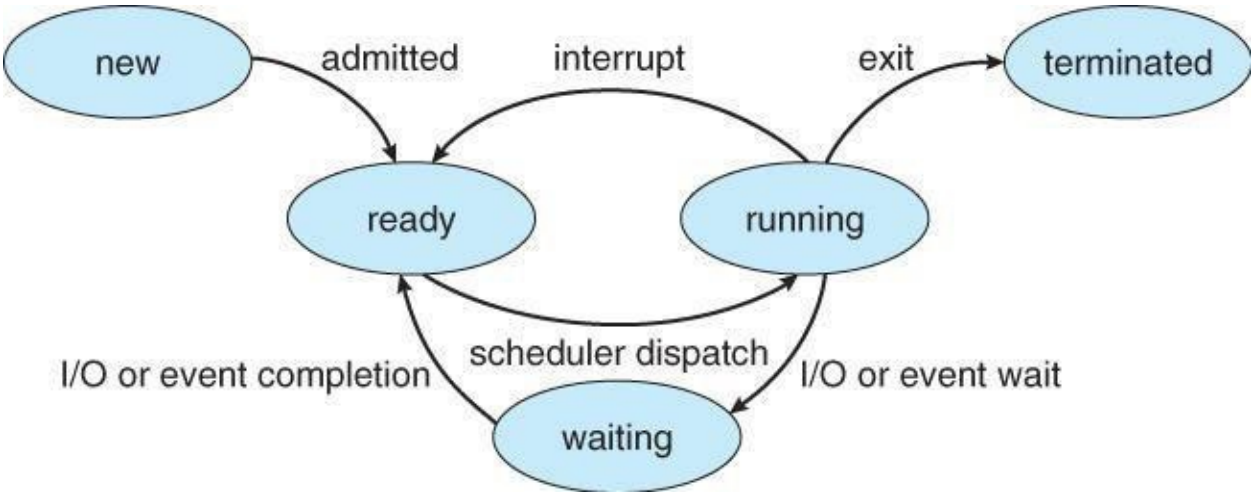
Process States

When a process executed, it changes the state, generally the state of process is determined by the current activity of the process. Each process may be in one of the following states:

1. New : The process is being created.
2. Running : The process is being executed.
3. Waiting : The process is waiting for some event to occur.
4. Ready : The process is waiting to be assigned to a processor.
5. Terminated : The Process has finished execution.

Only one process can be running in any processor at any time, But many process may be in ready and waiting states. The ready processes are loaded into a “ready queue”.

Diagram of process state



a) New -> Ready : OS creates process and prepares the process to be executed, then OS moved the process into ready queue.

b) Ready-> Running : OS selects one of the Jobs from ready Queue and move them from ready to Running.

c) Running-> Terminated : When the Execution of a process has Completed, OS terminates that process from running state. Sometimes OS terminates the process for some other reasons including Time exceeded, memory unavailable, access violation, protection Error, I/O failure and soon.

d) Running-> Ready : When the time slot of the processor expired (or) If the

processor receive an interrupt signal, the OS shifted Running -> Ready State.

e) Running -> Waiting : A process is put into the waiting state, if the process need an event occur (or) an I/O Device require.

f) Waiting->Ready : A process in the waiting state is moved to ready state when the event for which it has been Completed.

Process Control Block:

- Each process is represented in the operating System by a Process Control Block.
  It is also called Task Control Block.
  It contains many pieces of information associated with a specific Process.

Process State
Program Counter
CPU Registers
CPU Scheduling Information
Memory – Management Information
Accounting Information
I/O Status Information

Process Control Block

1. **Process State** : The State may be new, ready, running, and waiting, Terminated…
2. **Program Counter** : indicates the Address of the next Instruction to be executed.

3. **CPU registers** : registers include accumulators, stack pointers, General purpose Registers….

4. **CPU-Scheduling Info** : includes a process pointer, pointers to scheduling Queues, other scheduling parameter set.

5. **Memory management Info**: includes page tables, segmentation tables, value of base and limit registers.

6. **Accounting Information:** includes amount of CPU used, time limits, Jobs (or)Process numbers.

7. **I/O Status Information**: Includes the list of I/O Devices Allocated to the processes, list of open files.

NB: Context Switch: Assume, main memory contains more than one process. If cpu is executing a process, if time expires or if a high priority process enters into main memory, then the scheduler saves information about current process in the PCB and switches to execute the another process. The concept of moving CPU by scheduler from one process to other process is known as context switch.

WINDOW INSTALLATION:-