LINKEDIN

GMAIL

TELEGRAM

INSTAGRAM

GITHUB

Operating System (OS) Tutorial

🐛 Report Bug . ❓ Ask Question . 👍 Like the repo . 👎 Dislike the repo

> **Note**
>
> 📣 You can help to improve this repo, by giving me a ⭐**star**⭐ and ❤️**following me**❤️

Contents

Why Study Operating Systems?

- **Modern computer** consists of one or more **processors**, some **main memory**, **disks**, **printers**, a **keyboard**, a **mouse**, a **display**, **network interfaces**, and various other **input/output devices**. All in all, a **complex system**. If every application programmer had to understand how all these things work in detail, no code would ever get written. Furthermore, managing all these components and using them optimally is an **exceedingly challenging job**. For this reason, computers are equipped with a layer of **software** called the **operating system**.

- Although there are many practitioners of computer science, only a small percentage of them will be involved in the creation or modification of an operating system.

- Simply because, as almost all code runs on top of an operating system, knowledge of how operating systems work is crucial to proper, efficient, effective, and secure programming.

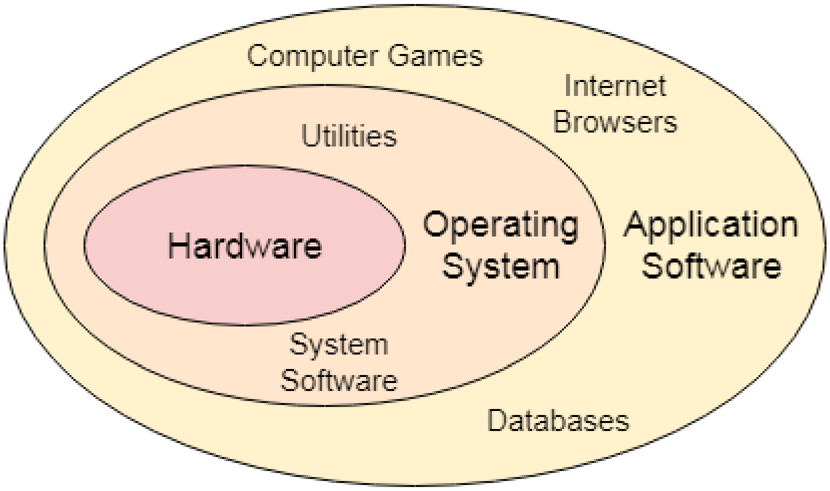
- Understanding the fundamentals of operating systems, how they drive computer hardware, and what they provide to applications is not only essential to those who program them but also highly useful to those who write programs on them and use them.

BACK TO UP

Operating System Definition and Function

- In the **Computer System** (comprises of **Hardware** and **Software**), **Hardware** can only understand **machine code** (in the form of **0** and **1**) which doesn't make any sense to a naive user.

- We need a system which can act as an **intermediary** and manage all the **processes** and **resources** present in the system.



- An **Operating System** can be defined as an **interface between user and hardware**. It is **responsible** for the **execution** of all the **processes**, **Resource Allocation**, **CPU management**, **File Management** and many other tasks.

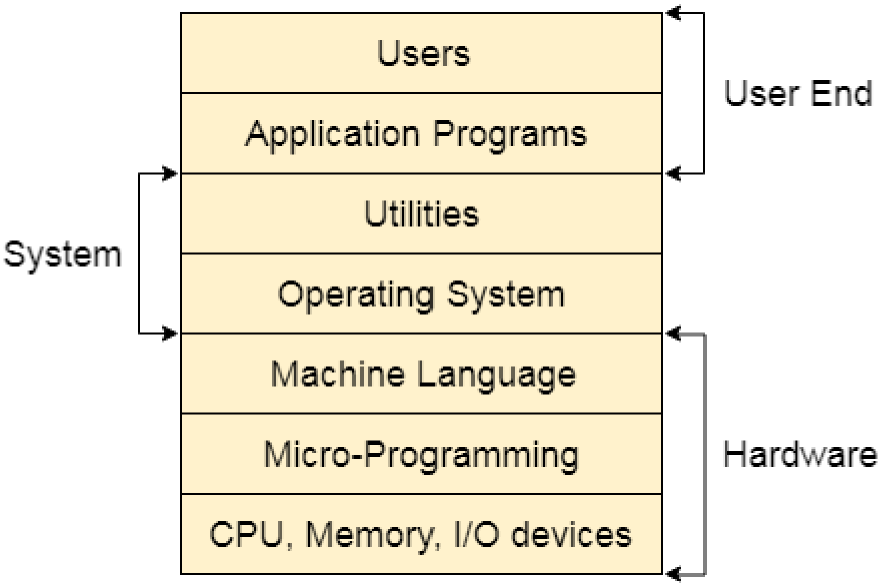
- The **purpose** of an operating system is to **provide an environment** in which a user can **execute programs** in **convenient and efficient manner**.

BACK TO UP

Structure of a Computer System

- ## A Computer System consists of:
    - ### Users
        - People who are using the computer.
    - ### Application Programs
        - Compilers, Databases, Games, Video player, Browsers, etc.
    - ### System Programs
        - Shells, Editors, Compilers, etc.

- **Operating System**
    - A special program which acts as an interface between user and hardware.
- **Hardware**
    - CPU, Disks, Memory, I/O Devices ,etc.



BACK TO UP

What Does an Operating system Do?

1. Process Management

2. Process Synchronization

3. Memory Management

4. CPU Scheduling

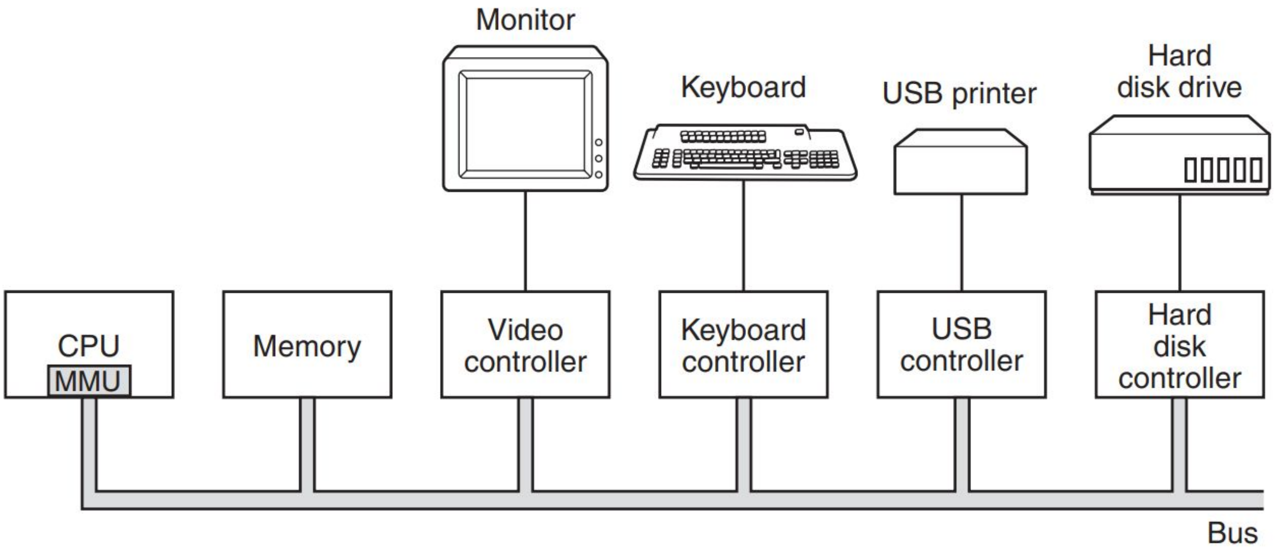
5. File Management

6.Security

BACK TO UP

Computer Hardware Review

-
  - **Operating system** is intimately tied to the **hardware** of the computer it runs on. It **extends** the **computer’s instruction set** and **manages its resources**.



- The **CPU**, **memory**, and **I/O devices** are all connected by a **system bus** and communicate with one another over it

1. Processors

- The **brain** of the **computer** is the **CPU (Central Processing Unit)**. It **fetches instructions** from **memory** and **executes** them.

- The basic **cycle** of every **CPU** is to **fetch** the **first instruction** from **memory**, **decode** it to **determine its type and operands**, **execute** it, and then **fetch**, **decode**, and **execute subsequent instructions**.

- The cycle is repeated **until** the **program finishes**. In this way, programs are **carried out**.

- All **CPUs contain** some **registers** inside to **hold key variables and temporary results**.

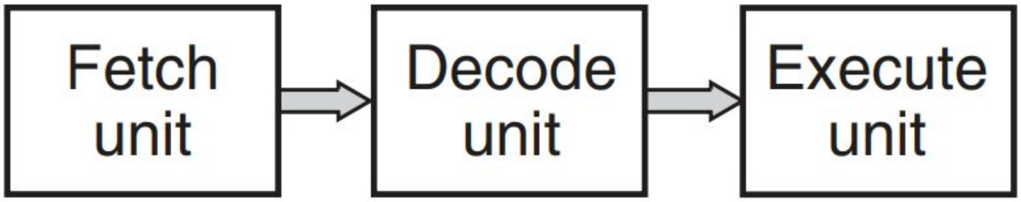
- Most computers have several special registers that are visible to the programmer. One of these is the **program counter (PC)**, which contains the **memory address of the next instruction** to be **fetched**.

- **Stack pointer (SP)**, which points to the **top of the current stack in memory**.

- **PSW (Program Status Word)**, this register contains the **condition code bits**, which are **set by comparison instructions**, the **CPU priority**, the **mode (user or kernel)**, and **various other control bits**.

- The **PSW** plays an **important role** in **system calls** and **I/O**.

- To **improve performance**, **CPU designers** have long **abandoned** the simple model of **fetching**, **decoding**, and **executing one instruction at a time**. Many **modern CPUs** have **facilities for executing more than one instruction** at the same time.
    - For example, a **CPU** might have **separate fetch**, **decode**, and **execute units**, Such an organization is called a **pipeline**.



- Even **more advanced than** a **pipeline design** is a **superscalar CPU**. In this design, **multiple execution units** are present
    - For example, one for integer arithmetic, one for floating-point arithmetic, and one for Boolean operations. Two or more instructions are fetched at once, decoded, and dumped into a holding buffer until they can be executed.

BACK TO UP

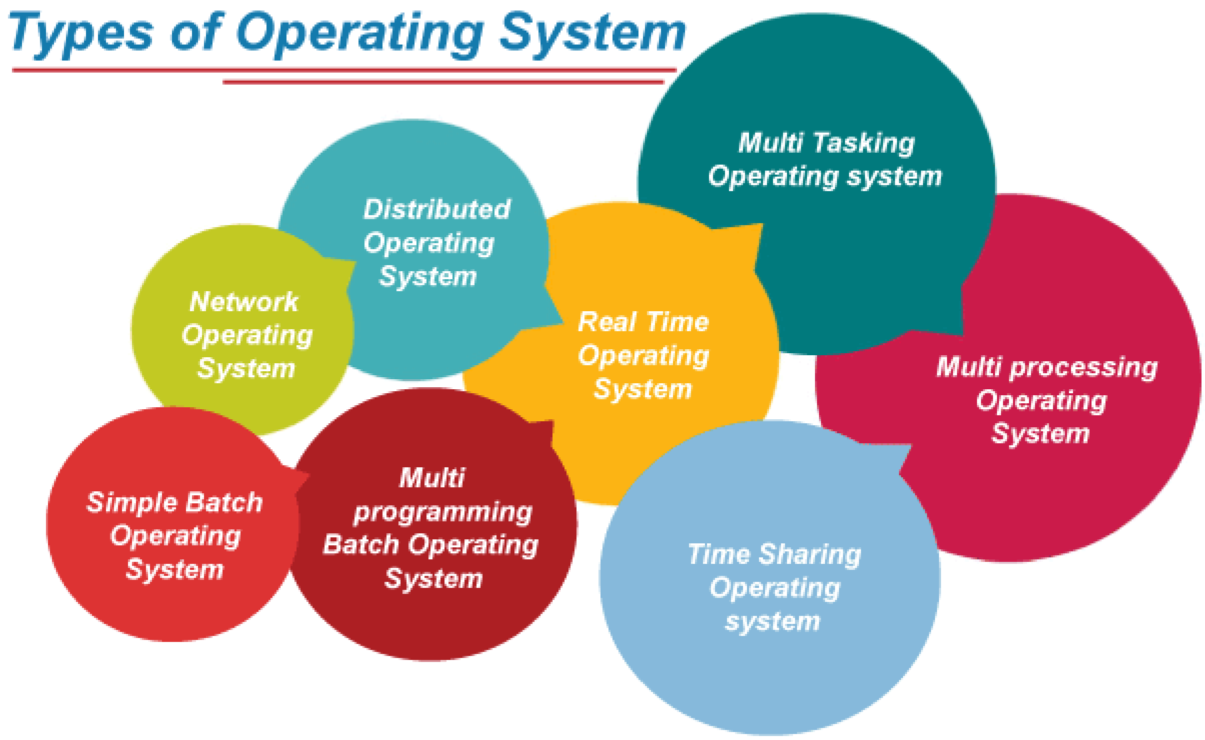
Operating Systems - History

- **1945 – 1955 : The First Generation, Vacuum Tubes and Plugboards**
- **1955 – 1965 : The Second Generation, Transistors and Batch Systems**
- **1965 - 1980 : The Third Generation, ICs and Multiprogramming**
- **1980 - Present : The Fourth Generation, Personal Computers**
- **1990 - Present : The Fifth Generation, Moblie Computets**

BACK TO UP

Types of Operating Systems

- **Operating System** is a well-organized collection of programs that manages the computer hardware. It is a type of **system software** that is responsible for the smooth functioning of the computer system.


Types of Operating System
Multi Tasking Operating system
Distributed Operating System
Network Operating System
Real Time Operating System
Multi processing Operating System
Simple Batch Operating System
Multi programming Batch Operating System
Time Sharing Operating system

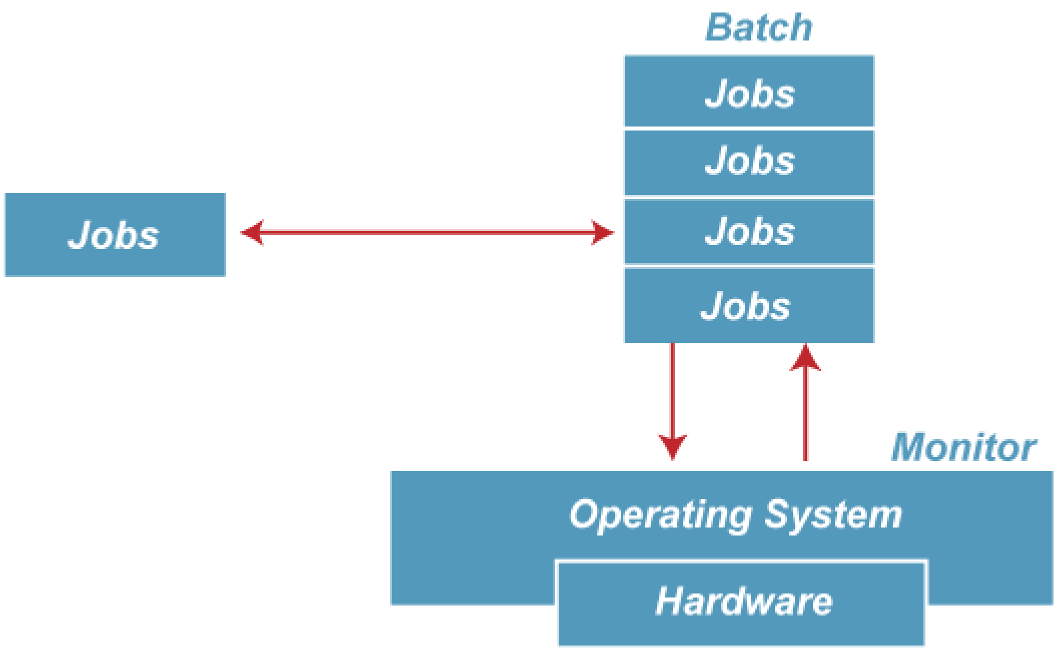
BACK TO UP

Batch Operating System

- In the **1970s**, **Batch processing** was very popular. In this technique, **similar types of jobs** were **batched together** and executed in time.

- In **Batch operating system**, **access** is given to **more than one person**; they submit their respective jobs to the system for the execution.

- The system put all of the jobs in a **queue** on the basis of **first come**, **first serve** and then executes the jobs **one by one**. The users collect their respective output when **all the jobs** get executed.



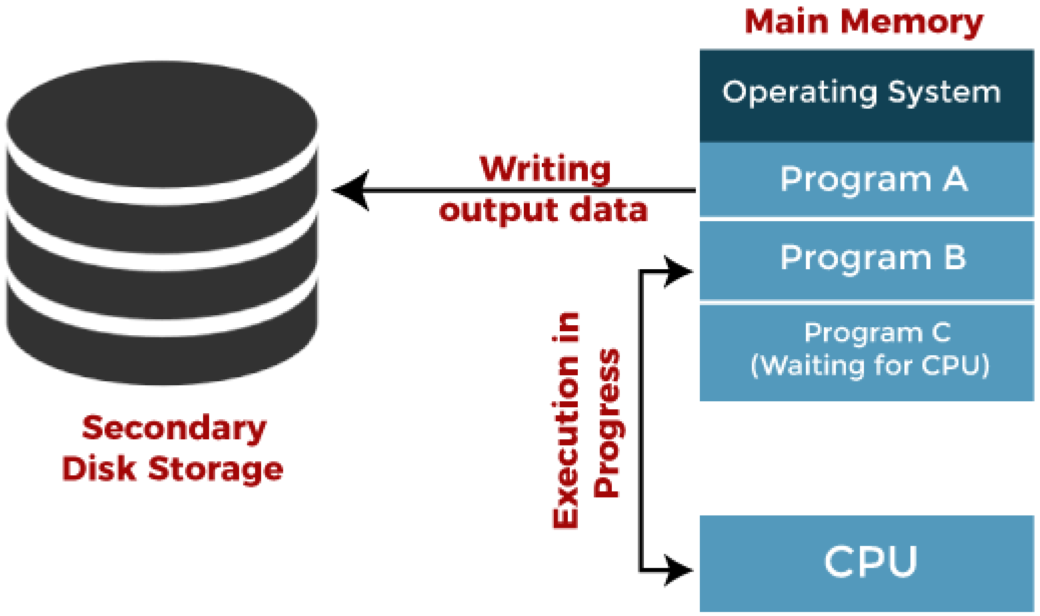
- The **purpose** of this operating system was mainly to **transfer control** from one job to another as soon as the job was completed.

BACK TO UP

Multiprogramming Operating System

- **Multiprogramming** is an **extension** to **batch processing** where the **CPU** is always **kept busy**. **Each process** needs two types of **system time**:
  1. **CPU time**
  2. **IO time**

- In a **multiprogramming environment**, when a process does its I/O, The CPU can start the execution of other processes. Therefore, multiprogramming improves the efficiency of the system.

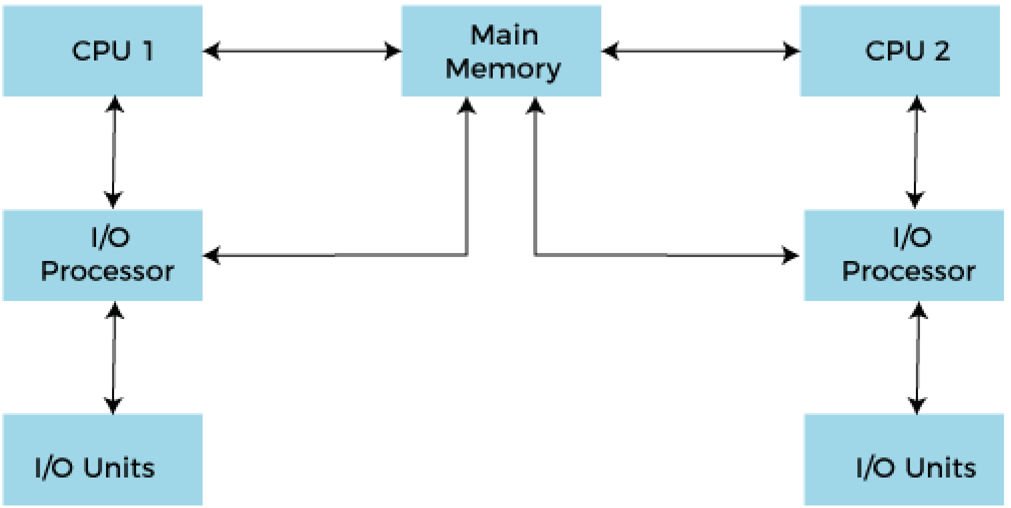


Jobs in multiprogramming system

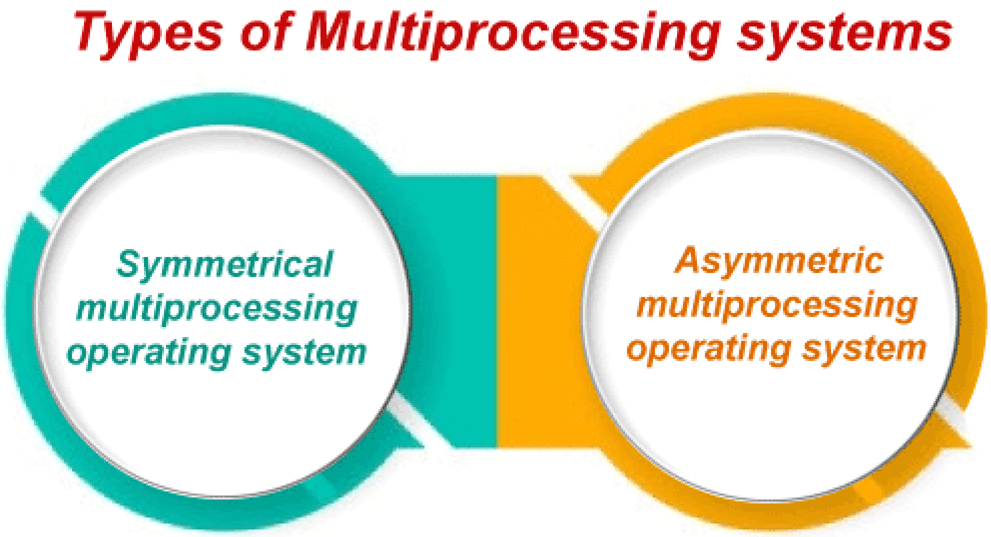
BACK TO UP

Multiprocessing Operating System

- In **Multiprocessing**, **Parallel computing** is achieved. There are **more than one processors** present in the system which can execute more than one process at the same time. This will increase the throughput of the system.



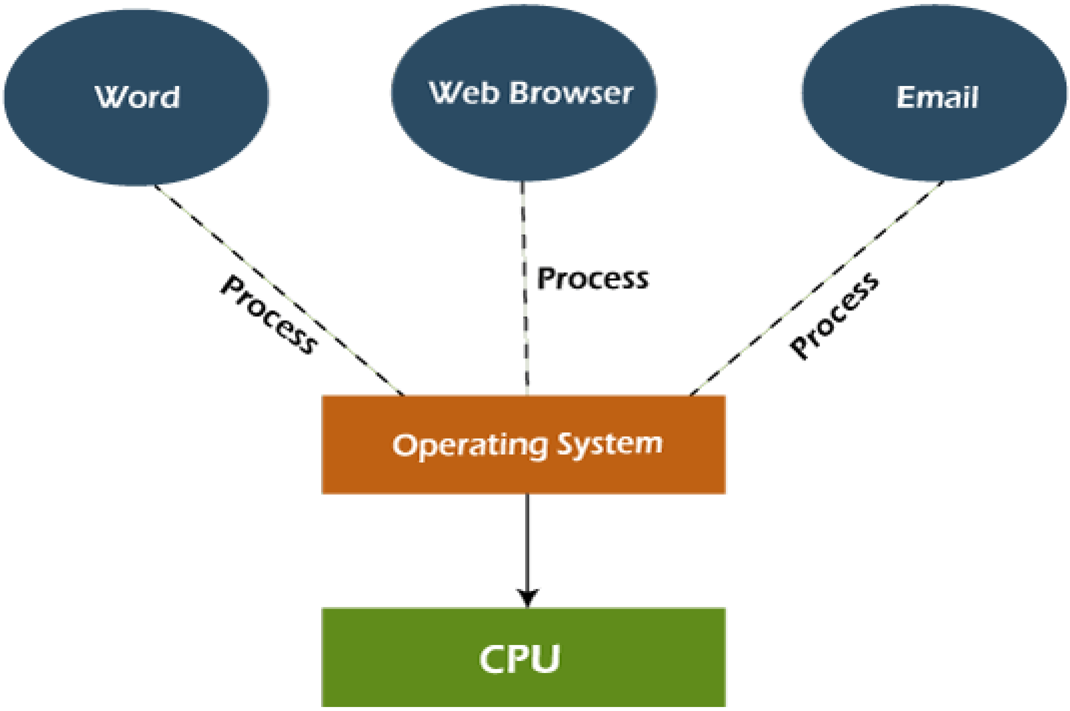
Working of Multiprocessor System

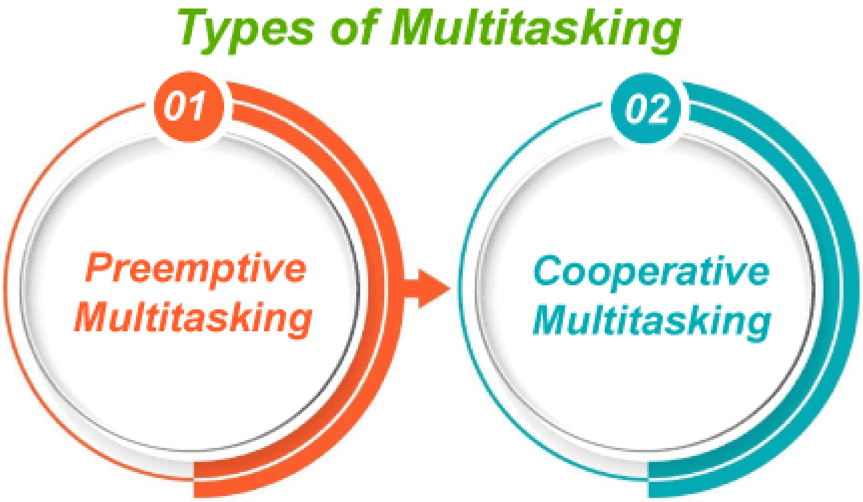

Types of Multiprocessing systems
Symmetrical multiprocessing operating system
Asymmetric multiprocessing operating system

BACK TO UP

Multitasking Operating System

- The **multitasking operating system** is a **logical extension of a multiprogramming system** that enables **multiple programs** simultaneously. It allows a user to perform more than one computer task at the same time.

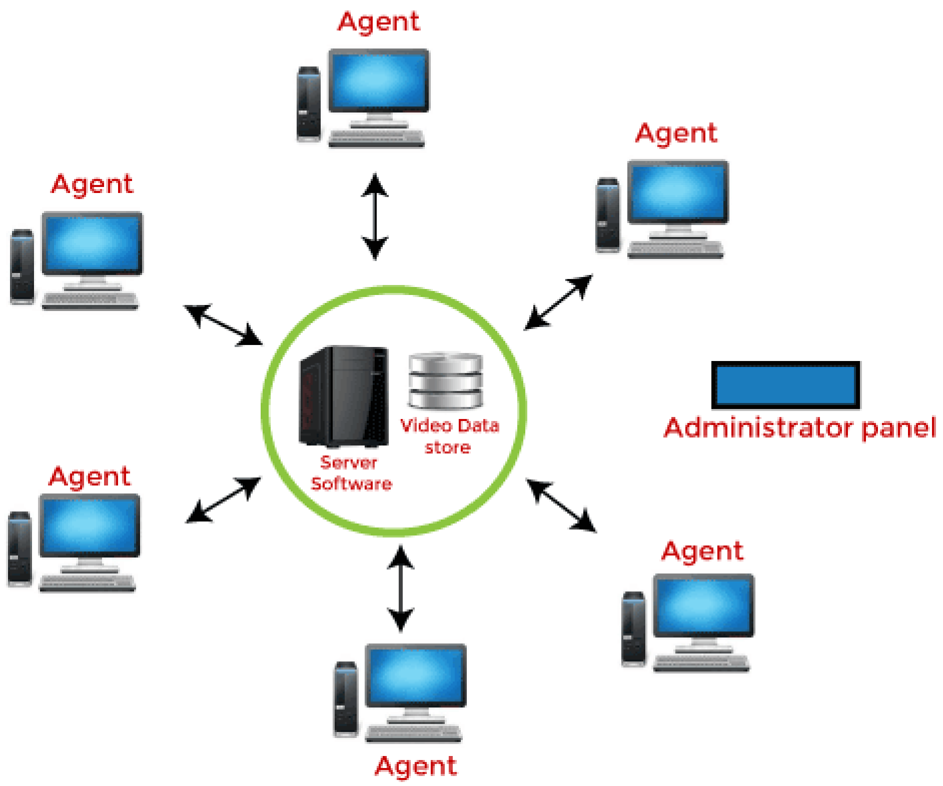




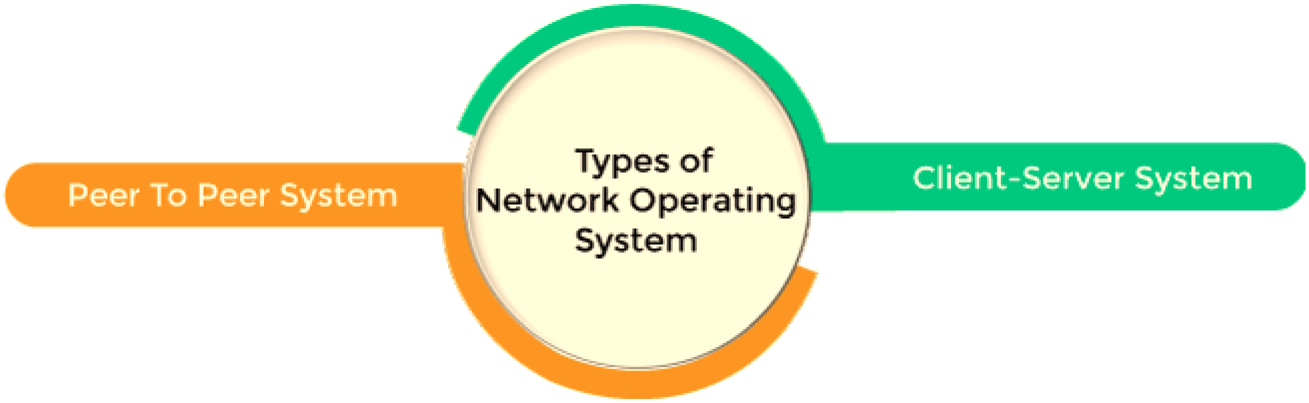
BACK TO UP

Network Operating Systems

- An **Operating system**, which includes **software** and **associated protocols** to communicate with **other computers** via a **network** conveniently and cost-effectively, is called **Network Operating System**.



Network Operating Systems



- In this type of operating system, **network traffic reduces** due to the **division between clients and the server**.

- This type of system is **less expensive** to **set up** and **maintain**.

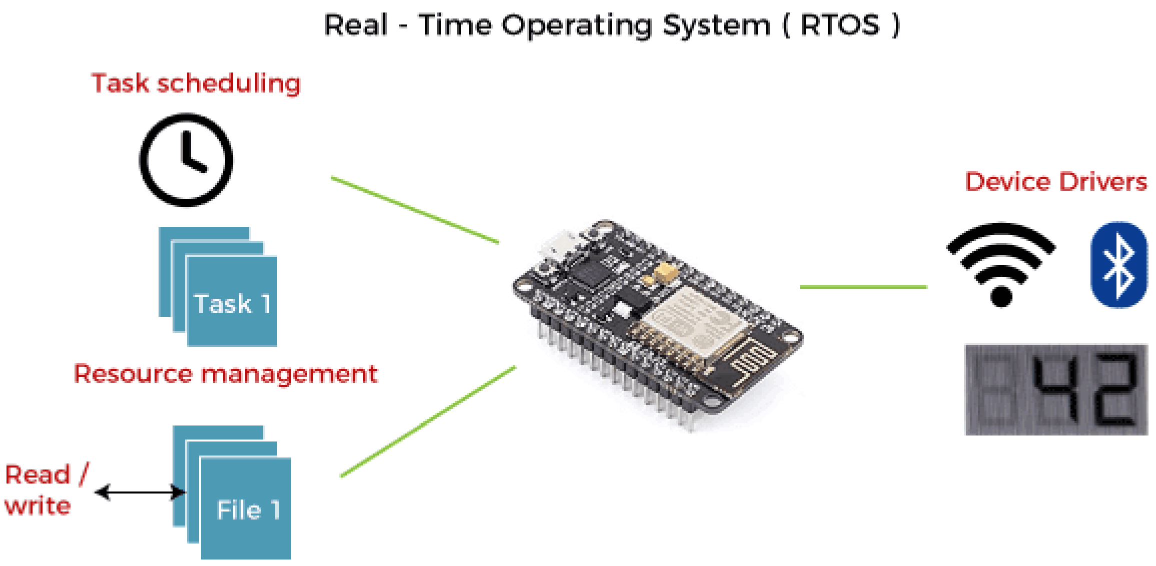
- In this type of operating system, the **failure of any node** in a system affects the whole system.

- **Security and performance are important issues**. So trained network administrators are required for network administration.

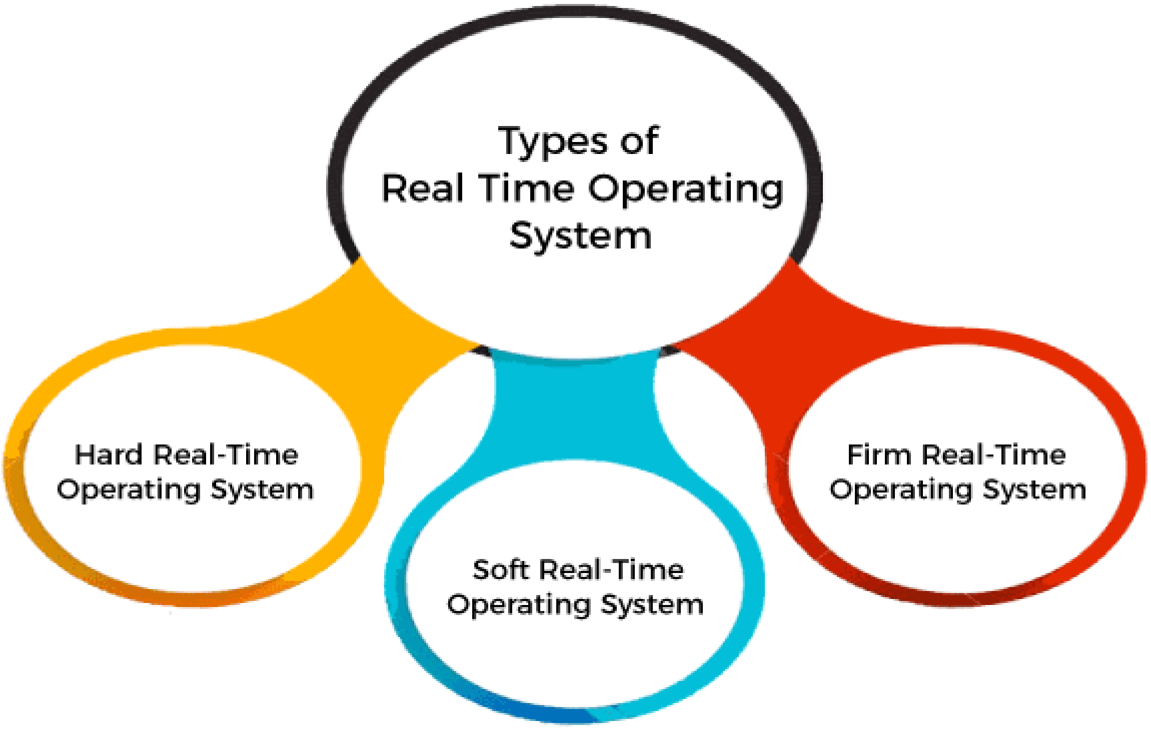
BACK TO UP

Real Time Operating System

- In **Real-Time Systems**, each job **carries a certain deadline** within which the job is supposed to be **completed**, **otherwise**, the huge loss will be there, or even if the result is produced, it will be completely useless.



- The **Application of a Real-Time system** exists in the case of **military applications**, if you want to drop a missile, then the missile is supposed to be dropped with a certain precision.

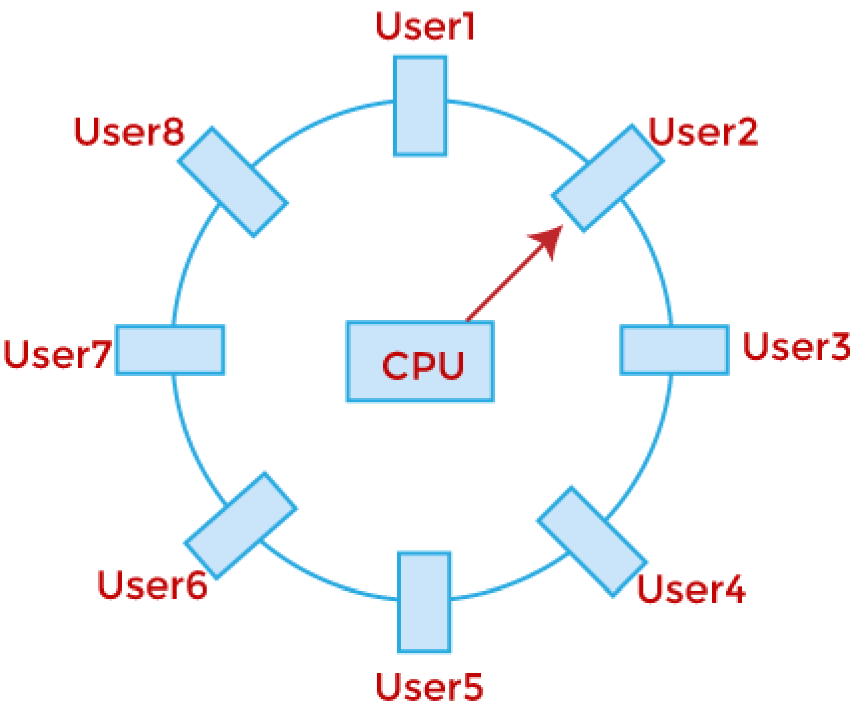


- Easy to layout, develop and execute real-time applications under the real-time operating system.
- In a Real-time operating system, the maximum utilization of devices and systems.
- Real-time operating systems are very costly to develop.
- Real-time operating systems are very complex and can consume critical CPU cycles.

BACK TO UP

Time-Sharing Operating System

- In the **Time Sharing operating system**, computer resources are allocated in a time-dependent fashion to several programs simultaneously. Thus it helps to provide a large number of user's direct access to the main computer.
- It is a **logical extension of multiprogramming**. In time-sharing, the CPU is switched among multiple programs given by different users on a scheduled basis.



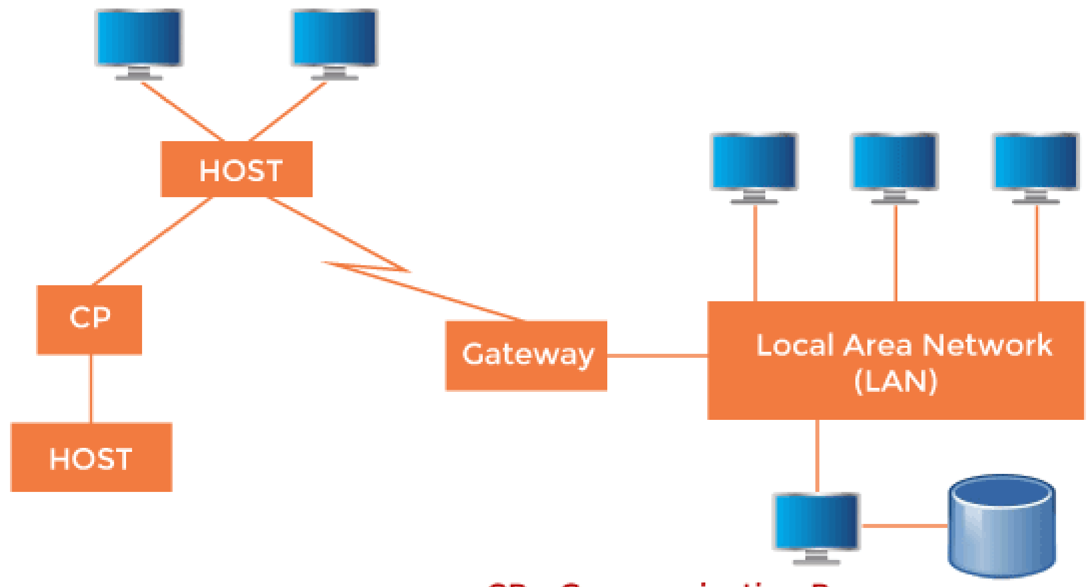
Timesharing in case of 8 users

- A time-sharing operating system allows many users to be served simultaneously, so sophisticated CPU scheduling schemes and Input/output management are required.
- Time-sharing operating systems are very difficult and expensive to build.
- The time-sharing operating system provides effective utilization and sharing of resources.
- This system reduces CPU idle and response time.
- Data transmission rates are very high in comparison to other methods.
- Security and integrity of user programs loaded in memory and data need to be maintained as many users access the system at the same time.

BACK TO UP

Distributed Operating System

- The Distributed Operating system is not installed on a single machine, it is divided into parts, and these parts are loaded on different machines. A part of the distributed Operating system is installed on each machine to make their communication possible. Distributed Operating systems are much more complex, large, and sophisticated than Network operating systems because they also have to take care of varying networking protocols.



CP - Communication Processors

A Typical View of a Distributed System

- The distributed operating system provides sharing of resources.
- This type of system is fault-tolerant.
- Protocol overhead can dominate computation cost.

BACK TO UP