Introduction of Operating System

An operating system acts as an intermediary between the user of a computer and computer hardware. The purpose of an operating system is to provide an environment in which a user can execute programs conveniently and efficiently.

An operating system is a software that manages computer hardware. The hardware must provide appropriate mechanisms to ensure the correct operation of the computer system and to prevent user programs from interfering with the proper operation of the system.

**Operating System –** Definition: 

* An operating system is a program that controls the execution of application programs and acts as an interface between the user of a computer and the computer hardware.
* A more common definition is that the operating system is the one program running at all times on the computer (usually called the kernel), with all else being application programs.
* An operating system is concerned with the allocation of resources and services, such as memory, processors, devices, and information. The operating system correspondingly includes programs to manage these resources, such as a traffic controller, a scheduler, a memory management module, I/O programs, and a file system.

**Features of Operating system –** Operating system has the following features: 

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

**Major Functionalities of Operating System:**

* **Resource Management:**When parallel accessing happens in the OS means when multiple users are accessing the system the OS works as Resource Manager, Its responsibility is to provide hardware to the user. It decreases the load in the system.
* **Process Management:**It includes various tasks like **scheduling and termination**of the process. It is done with the help of **CPU Scheduling** algorithms.
* **Storage Management:**The **file system** mechanism used for the management of the storage. **NIFS**, **CFS**, **CIFS**, **NFS**, etc. are some file systems. All the data is stored in various tracks of Hard disks that are all managed by the storage manager. It included **Hard Disk**.
* **Memory Management:**Refers to the management of primary memory. The operating system has to keep track of how much memory has been used and by whom. It has to decide which process needs memory space and how much. OS also has to allocate and deallocate the memory space.
* **Security/Privacy Management:**Privacy is also provided by the Operating system by means of passwords so that unauthorized applications can’t access programs or data. For example, Windows uses **Kerberos** authentication to prevent unauthorized access to data.

The process operating system as User Interface:

1. User
2. System and application programs
3. Operating system
4. Hardware

Every general-purpose computer consists of hardware, an operating system(s), system programs, and application programs. The hardware consists of memory, CPU, ALU, I/O devices, peripheral devices, and storage devices. The system program consists of compilers, loaders, editors, OS, etc. The application program consists of business programs, and database programs.

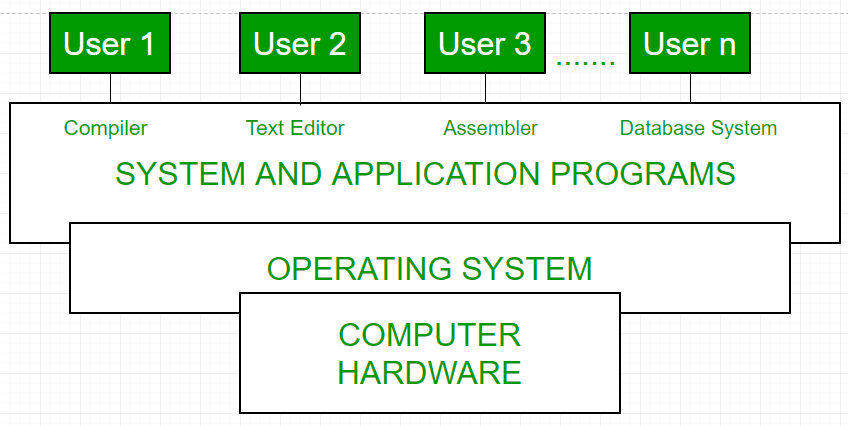


Fig1: Conceptual view of a computer system

Every computer must have an operating system to run other programs. The operating system coordinates the use of the hardware among the various system programs and application programs for various users. It simply provides an environment within which other programs can do useful work.

The operating system is a set of special programs that run on a computer system that allows it to work properly. It performs basic tasks such as recognizing input from the keyboard, keeping track of files and directories on the disk, sending output to the display screen, and controlling peripheral devices.   
OS is designed to serve two basic purposes: 

1. It controls the allocation and use of the computing System’s resources among the various user and tasks.
2. It provides an interface between the computer hardware and the programmer that simplifies and makes it feasible for coding and debugging of application programs.

The Operating system must support the following tasks. The tasks are:  

1. Provides the facilities to create and modify of programs and data files using an editor.
2. Access to the compiler for translating the user program from high-level language to machine language.
3. Provide a loader program to move the compiled program code to the computer’s memory for execution.
4. Provide routines that handle the details of I/O programming.

**I/O System Management –**   
The module that keeps track of the status of devices is called the I/O traffic controller. Each I/O device has a device handler that resides in a separate process associated with that device.   
The I/O subsystem consists of 

* A memory Management component that includes buffering caching and spooling.
* A general device driver interface.

Drivers for specific hardware devices.

**Assembler –**   
The input to an assembler is an assembly language program. The output is an object program plus information that enables the loader to prepare the object program for execution. At one time, the computer programmer had at his disposal a basic machine that interpreted, through hardware, certain fundamental instructions. He would program this computer by writing a series of ones and Zeros (Machine language), and place them into the memory of the machine.

**Compiler and Interpreter –**   
The High-level languages- examples are FORTRAN, COBOL, ALGOL, and PL/I are processed by compilers and interpreters. A compiler is a program that accepts a source program in a “high-level language “and produces a corresponding object program. An interpreter is a program that appears to execute a source program as if it was machine language. The same name (FORTRAN, COBOL, etc.) is often used to designate both a compiler and its associated language.

**Loader –**   
A Loader is a routine that loads an object program and prepares it for execution. There are various loading schemes: absolute, relocating, and direct-linking. In general, the loader must load, relocate and link the object program. The loader is a program that places programs into memory and prepares them for execution. In a simple loading scheme, the assembler outputs the machine language translation of a program on a secondary device and a loader places it in the core. The loader places into memory the machine language version of the user’s program and transfers control to it. Since the loader program is much smaller than the assembler, those make more core available to the user’s program.

**History of Operating system –**   
The operating system has been evolving through the years. The following table shows the history of OS. 

|  |  |  |  |
| --- | --- | --- | --- |
| Generation | Year | Electronic device used | Types of OS Devices |
| First | 1945-55 | Vacuum Tubes | Plug Boards |
| Second | 1955-65 | Transistors | Batch Systems |
| Third | 1965-80 | Integrated Circuits(IC) | Multiprogramming |
| Fourth | Since 1980 | Large Scale Integration | PC |

[**Types of Operating Systems**](https://www.geeksforgeeks.org/operating-system-types-operating-systems-awaiting-author/)**–**

* Batch Operating System- Sequence of jobs in a program on a computer without manual interventions.
* Time-sharing operating System- allows many users to share computer resources. (Max utilization of the resources).
* Distributed operating System- Manages a group of different computers and makes appear to be a single computer.
* Network operating system- computers running in different operating systems can participate in a common network (It is used for security purposes).
* Real-time operating system – used where any or all the jobs have strict time constraints.

Examples of Operating Systems are –

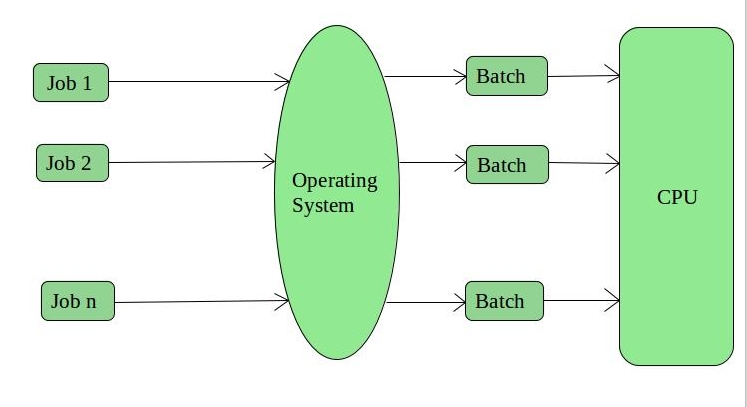
* Windows (GUI-based, PC)
* GNU/Linux (Personal, Workstations, ISP, File and print server, Three-tier client/Server)
* macOS (Macintosh), used for Apple’s personal computers and workstations (MacBook, iMac).
* Android (Google’s Operating System for smartphones/tablets/smartwatches)
* iOS (Apple’s OS for iPhone, iPad, and iPod Touch)

Types of Operating System

An Operating System performs all the basic tasks like managing files, processes, and memory. Thus operating system acts as the manager of all the resources, i.e. **resource manager**. Thus, the operating system becomes an interface between user and machine.

**Types of Operating Systems:** Some widely used operating systems are as follows-

**1. Batch Operating System –**   
This type of operating system does not interact with the computer directly. There is an operator which takes similar jobs having the same requirement and group them into batches. It is the responsibility of the operator to sort jobs with similar needs.



**Advantages of Batch Operating System:**

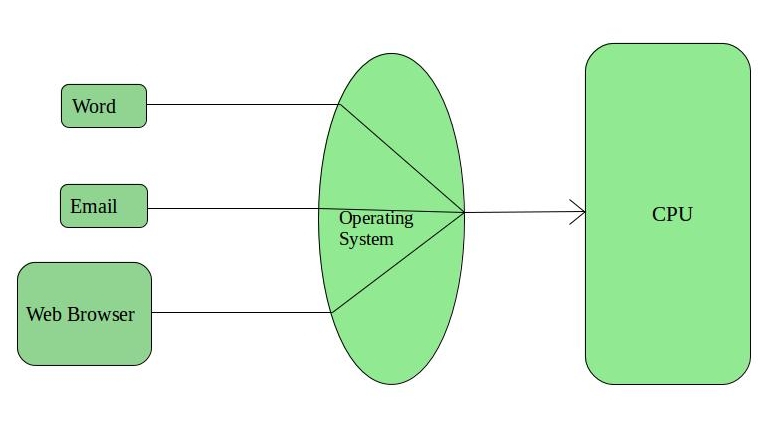
* It is very difficult to guess or know the time required for any job to complete. Processors of the batch systems know how long the job would be when it is in queue
* Multiple users can share the batch systems
* The idle time for the batch system is very less
* It is easy to manage large work repeatedly in batch systems

**Disadvantages of Batch Operating System:**

* The computer operators should be well known with batch systems
* Batch systems are hard to debug
* It is sometimes costly
* The other jobs will have to wait for an unknown time if any job fails

**Examples of Batch based Operating System:** Payroll System, Bank Statements, etc.

**2. Time-Sharing Operating Systems –**   
Each task is given some time to execute so that all the tasks work smoothly. Each user gets the time of CPU as they use a single system. These systems are also known as Multitasking Systems. The task can be from a single user or different users also. The time that each task gets to execute is called quantum. After this time interval is over OS switches over to the next task.



**Advantages of Time-Sharing OS:**

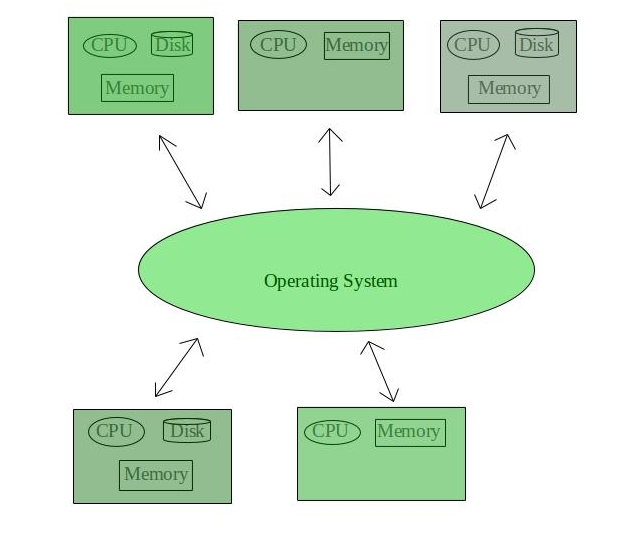
* Each task gets an equal opportunity
* Fewer chances of duplication of software
* CPU idle time can be reduced

**Disadvantages of Time-Sharing OS:**

* Reliability problem
* One must have to take care of the security and integrity of user programs and data
* Data communication problem

**Examples of Time-Sharing OSs are:** Multics, Unix, etc.

**3. Distributed Operating System –**   
These types of the operating system is a recent advancement in the world of computer technology and are being widely accepted all over the world and, that too, with a great pace. Various autonomous interconnected computers communicate with each other using a shared communication network. Independent systems possess their own memory unit and CPU. These are referred to as **loosely coupled systems** or distributed systems. These system’s processors differ in size and function. The major benefit of working with these types of the operating system is that it is always possible that one user can access the files or software which are not actually present on his system but some other system connected within this network i.e., remote access is enabled within the devices connected in that network. 



**Advantages of Distributed Operating System:**

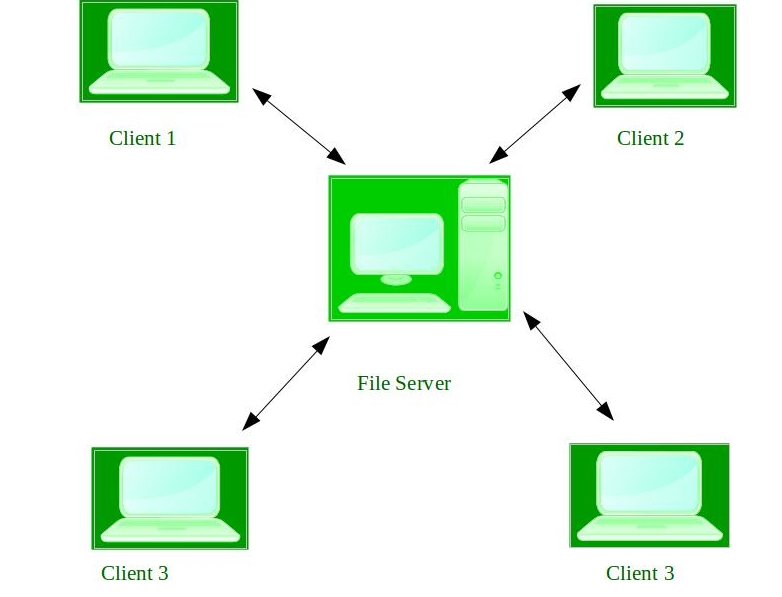
* Failure of one will not affect the other network communication, as all systems are independent from each other
* Electronic mail increases the data exchange speed
* Since resources are being shared, computation is highly fast and durable
* Load on host computer reduces
* These systems are easily scalable as many systems can be easily added to the network
* Delay in data processing reduces

**Disadvantages of Distributed Operating System:**

* Failure of the main network will stop the entire communication
* To establish distributed systems the language which is used are not well defined yet
* These types of systems are not readily available as they are very expensive. Not only that the underlying software is highly complex and not understood well yet

**Examples of Distributed Operating System are-** LOCUS, etc.

**4. Network Operating System –**   
These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions. These types of operating systems allow shared access of files, printers, security, applications, and other networking functions over a small private network. One more important aspect of Network Operating Systems is that all the users are well aware of the underlying configuration, of all other users within the network, their individual connections, etc. and that’s why these computers are popularly known as **tightly coupled systems**.



**Advantages of Network Operating System:**

* Highly stable centralized servers
* Security concerns are handled through servers
* New technologies and hardware up-gradation are easily integrated into the system
* Server access is possible remotely from different locations and types of systems

**Disadvantages of Network Operating System:**

* Servers are costly
* User has to depend on a central location for most operations
* Maintenance and updates are required regularly

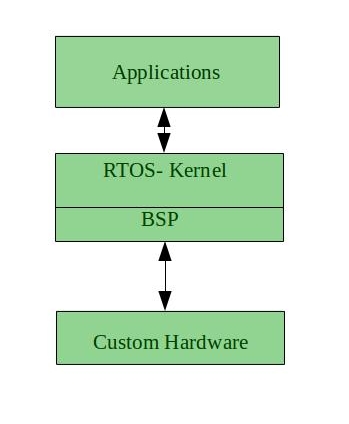
**Examples of Network Operating System are:** Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD, etc.

**5. Real-Time Operating System –**   
These types of OSs serve real-time systems. The time interval required to process and respond to inputs is very small. This time interval is called **response time**.

**Real-time systems** are used when there are time requirements that are very strict like missile systems, air traffic control systems, robots, etc.

**Two types of Real-Time Operating System which are as follows:**

* **Hard Real-Time Systems:**   
  These OSs are meant for applications where time constraints are very strict and even the shortest possible delay is not acceptable. These systems are built for saving life like automatic parachutes or airbags which are required to be readily available in case of any accident. Virtual memory is rarely found in these systems.
* **Soft Real-Time Systems:**   
  These OSs are for applications where for time-constraint is less strict.



**Advantages of RTOS:**

* **Maximum Consumption:** Maximum utilization of devices and system, thus more output from all the resources
* **Task Shifting:** The time assigned for shifting tasks in these systems are very less. For example, in older systems, it takes about 10 microseconds in shifting one task to another, and in the latest systems, it takes 3 microseconds.
* **Focus on Application:** Focus on running applications and less importance to applications which are in the queue.
* Real-time**operating system in**the **embedded system:** Since the size of programs are small, RTOS can also be used in embedded systems like in transport and others.
* **Error Free:** These types of systems are error-free.
* **Memory Allocation:** Memory allocation is best managed in these types of systems.

**Disadvantages of RTOS:**

* **Limited Tasks:** Very few tasks run at the same time and their concentration is very less on few applications to avoid errors.
* **Use heavy system resources:** Sometimes the system resources are not so good and they are expensive as well.
* **Complex Algorithms:** The algorithms are very complex and difficult for the designer to write on.
* **Device driver and interrupt signals:** It needs specific device drivers and interrupts signals to respond earliest to interrupts.
* **Thread Priority:** It is not good to set thread priority as these systems are very less prone to switching tasks.

**Examples of Real-Time Operating Systems are:** Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.