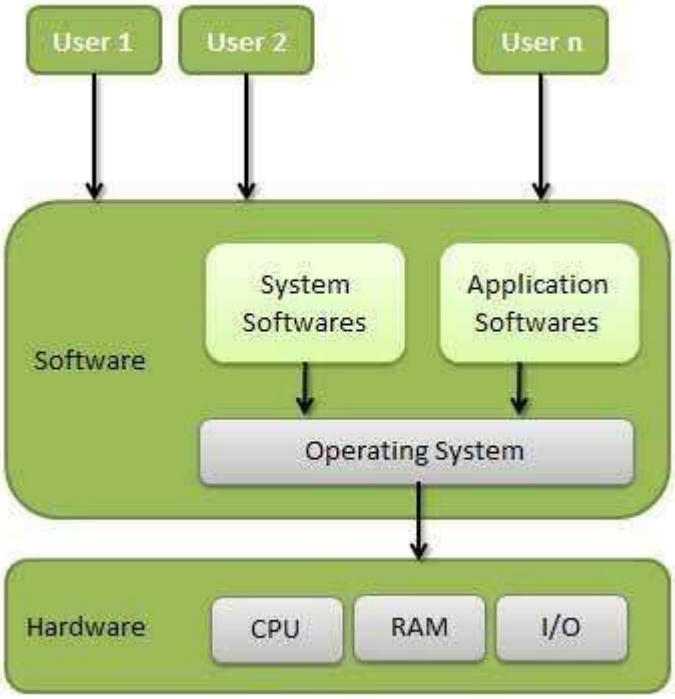
Unit- 1: Introduction of operating system

An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

Definition

An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.



Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.

Following are some of important functions of an 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

Applications of Operating System

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 softwares and users − Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

Types of operating system

Operating systems are there from the very first computer generation and they keep evolving with time. In this chapter, we will discuss some of the important types of operating systems which are most commonly used.

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 of Timesharing operating systems are as follows −

1. Provides the advantage of quick response.
2. Avoids duplication of software.
3. Reduces CPU idle time.

Disadvantages of Time-sharing operating systems are as follows −

1. Problem of reliability.
2. Question of security and integrity of user programs and data.
3. 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.

The advantages of distributed systems are as follows −

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

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. For 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.

1. **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.

2. **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. For example, multimedia, virtual reality, Advanced Scientific Projects like undersea exploration and planetary rovers.

3. **Batch processing**

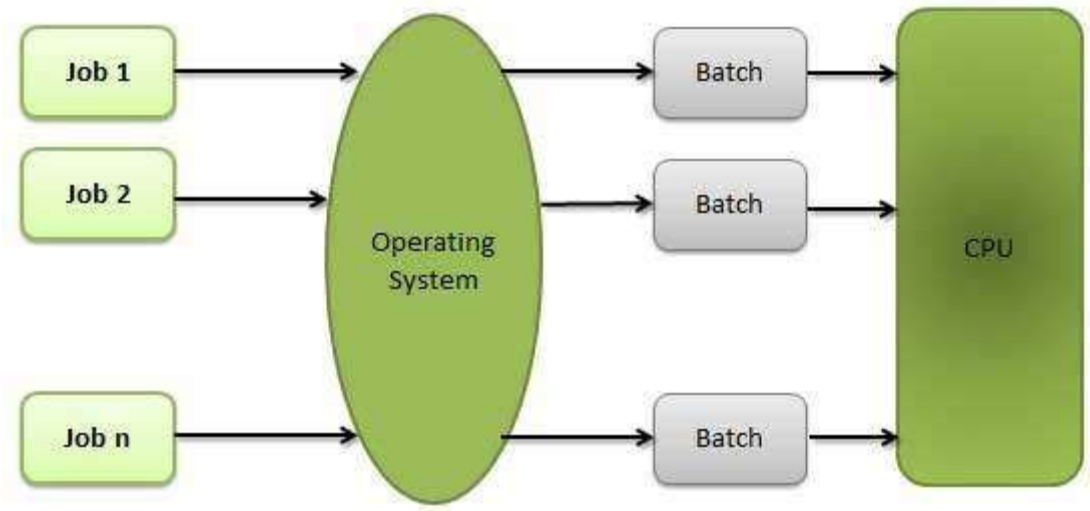
Batch processing is a technique in which an Operating System collects the programs and data together in a batch before processing starts. An operating system does the following activities related to batch processing –

The OS defines a job which has predefined sequence of commands, programs and data as a single unit.

The OS keeps a number a jobs in memory and executes them without any manual information.

Jobs are processed in the order of submission, i.e., first come first served fashion.

When a job completes its execution, its memory is released and the output for the job gets copied into an output spool for later printing or processing.



Advantages

1. Batch processing takes much of the work of the operator to the computer.
2. Increased performance as a new job get started as soon as the previous job is finished, without any manual intervention.

Disadvantages

1. Difficult to debug program.
2. A job could enter an infinite loop.

3. Due to lack of protection scheme, one batch job can affect pending jobs.

Multitasking

Multitasking is when multiple jobs are executed by the CPU simultaneously by switching between them. Switches occur so frequently that the users may interact with each program while it is running. An OS does the following activities related to multitasking −

The user gives instructions to the operating system or to a program directly, and receives an immediate response.

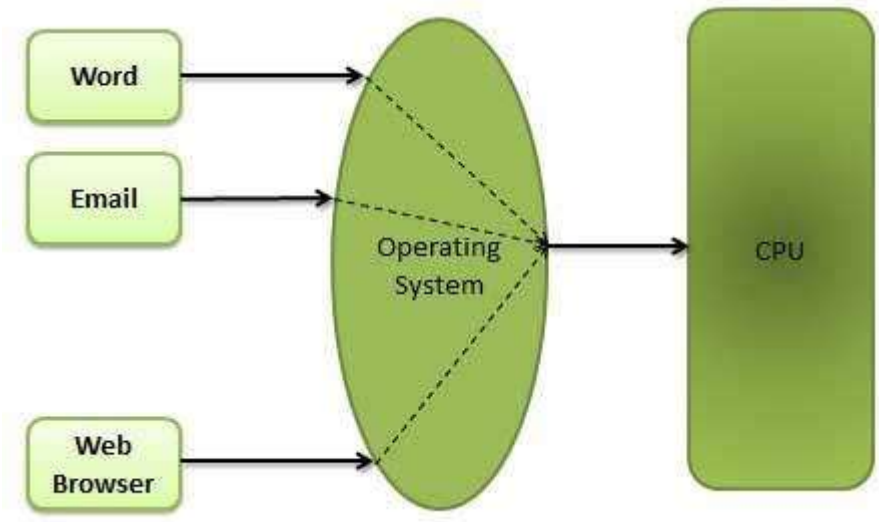
The OS handles multitasking in the way that it can handle multiple operations/executes multiple programs at a time.

Multitasking Operating Systems are also known as Time-sharing systems.

These Operating Systems were developed to provide interactive use of a computer system at a reasonable cost.

A time-shared operating system uses the concept of CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared CPU.

Each user has at least one separate program in memory.



A program that is loaded into memory and is executing is commonly referred to as a process.

When a process executes, it typically executes for only a very short time before it either finishes or needs to perform I/O.

Since interactive I/O typically runs at slower speeds, it may take a long time to complete. During this time, a CPU can be utilized by another process.

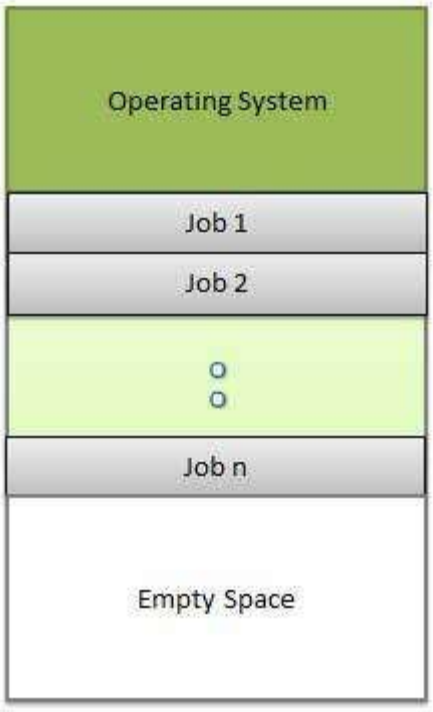
The operating system allows the users to share the computer simultaneously. Since each action or command in a time-shared system tends to be short, only a little CPU time is needed for each user.

As the system switches CPU rapidly from one user/program to the next, each user is given the impression that he/she has his/her own CPU, whereas actually one CPU is being shared among many users.

Multiprogramming

Sharing the processor, when two or more programs reside in memory at the same time, is referred as multiprogramming. Multiprogramming assumes a single shared processor. Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.

The following figure shows the memory layout for a multiprogramming system.



An OS does the following activities related to multiprogramming.

The operating system keeps several jobs in memory at a time.

This set of jobs is a subset of the jobs kept in the job pool.

The operating system picks and begins to execute one of the jobs in the memory.

Multiprogramming operating systems monitor the state of all active programs and system resources using memory management programs to ensures that the CPU is never idle, unless there are no jobs to process.

Advantages

1. High and efficient CPU utilization.
2. User feels that many programs are allotted CPU almost simultaneously.

Disadvantages

1. CPU scheduling is required.
2. To accommodate many jobs in memory, memory management is required.