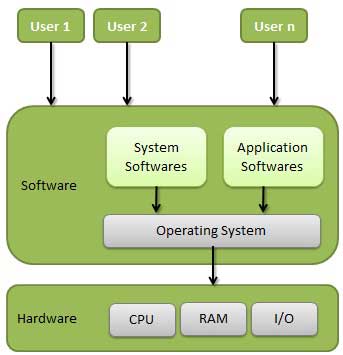
**Operating System and its Functions**

An operating System (OS) is an intermediary between users and computer hardware. It provides users an environment in which a user can execute programs conveniently and efficiently. In technical terms, It is a software which manages hardware. An operating System controls the allocation of resources and services such as memory, processors, devices and information.

**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.



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

**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 access directly by the CPU. So for a program to be executed, it must in the main memory. Operating System does the following activities for memory management.

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

**Processor Management**

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

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

**Device Management**

OS manages device communication via their respective drivers. Operating System does the following activities for device management.

* Keeps tracks of all devices. Program responsible for this task is known as the I/O controller.
* Decides which process gets the device when and for how much time.
* Allocates the device in the efficient way.
* 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. Operating System does the following activities for file management.

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

**Other Important Activities**

Following are some of the important activities that Operating System does.

* **Security** -- By means of password and similar other techniques, preventing 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 Systems**

Within the broad family of operating systems, there are generally four types, categorized based on the types of computers they control and the sort of applications they support. The categories are:

* **Real-time operating system** (RTOS) - Real-time operating systems are used to control machinery, scientific instruments and industrial systems. An RTOS typically has very little user-interface capability, and no end-user utilities, since the system will be a "sealed box" when delivered for use. A very important part of an RTOS is managing the resources of the computer so that a particular operation executes in precisely the same amount of time, every time it occurs. In a complex machine, having a part move more quickly just because system resources are available may be just as catastrophic as having it not move at all because the system is busy.
* **Single-user, single task** - As the name implies, this operating system is designed to manage the computer so that one user can effectively do one thing at a time. The Palm OS for Palm handheld computers is a good example of a modern single-user, single-task operating system.
* **Single-user, multi-tasking** - This is the type of operating system most people use on their desktop and laptop computers today. Microsoft's Windows and Apple's MacOS platforms are both examples of operating systems that will let a single user have several programs in operation at the same time. For example, it's entirely possible for a Windows user to be writing a note in a word processor while downloading a file from the Internet while printing the text of an e-mail message.
* **Multi-user** - A multi-user operating system allows many different users to take advantage of the computer's resources simultaneously. The operating system must make sure that the requirements of the various users are balanced, and that each of the programs they are using has sufficient and separate resources so that a problem with one user doesn't affect the entire community of users. Unix, VMS and mainframe operating systems, such as *MVS*, are examples of multi-user operating systems.

Operating systems are there from the very first computer generation. Operating systems keep evolving over the period of time. Following are few of the important types of operating system which are most commonly used.

**Batch operating system**

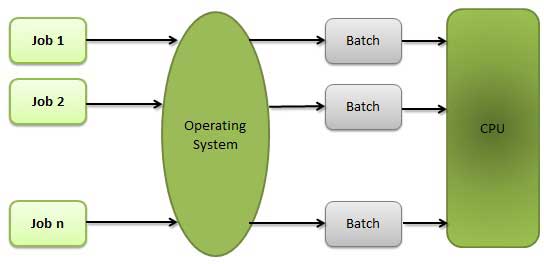
The users of batch operating system do not interact with the computer directly. Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group. Thus, the programmers left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

The problems with Batch Systems are following.

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

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

* OS defines a job which has predefined sequence of commands, programs and data as a single unit.
* OS keeps a number of 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 job completes its execution, its memory is released and the output for the job gets copied into an output for later printing or processing.



**Advantages**

* Batch processing takes much of the work of the operator to the computer.
* Increased performance as a new job gets started as soon as the previous job finished without any manual intervention.

**Disadvantages**

* Difficult to debug program.
* A job could enter an infinite loop.
* Due to lack of protection scheme, one batch job can affect pending jobs.

**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, objective is to maximize processor use, whereas in Time-Sharing Systems 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 receives an immediate response. For example, in a transaction processing, processor execute each user program in a short burst or quantum of computation. That is if n users are present, each user can get time quantum. When the user submits the command, the response time is in few seconds at most.

Computer systems that were designed primarily as batch systems have been modified to time-sharing systems.

Advantages of Timesharing operating systems are following

* Provide advantage of quick response.
* Avoids duplication of software.
* Reduces CPU idle time.

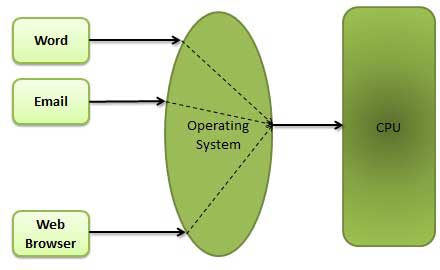
Disadvantages of Timesharing operating systems are following.

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

**Multitasking**

Multitasking refers to term where 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. Operating system 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.
* Operating System 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 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.

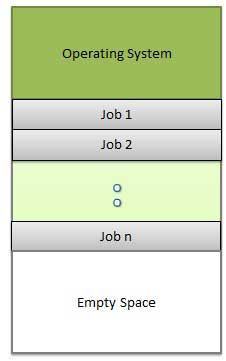


* A program that is loaded into memory and is executing is commonly referred to as a process.
* When a process executes, it typically executes for only a very short time before it either finishes or needs to perform I/O.
* Since interactive I/O typically runs at people speeds, it may take a long time to completed. During this time a CPU can be utilized by another process.
* 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**

When two or more programs are residing in memory at the same time, then sharing the processor is referred to the multiprogramming. Multiprogramming assumes a single shared processor. Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.

Following figure shows the memory layout for a multiprogramming system,



Operating system 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 job in the memory.
* Multiprogramming operating system monitors 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

**Advantages**

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

**Disadvantages**

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

**Real Time operating System**

Real time system is defines 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. Real time processing is always on line whereas on line system need not be real time. The time taken by the system to respond to an input and display of required updated information is termed as response time. So in this method response time is very less as compared to the 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. Real-time operating system has well-defined, fixed time constraints otherwise system will fail.For example Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, and home-applicance controllers, Air traffic control system 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 with data stored in ROM. In these systems virtual memory is almost never found.

**Soft real-time systems**

Soft real time systems are less restrictive. 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 etc.

**Note:**

**Process**

A process is a program in execution. The execution of a process must progress in a sequential fashion. Definition of process is following.

* A process is defined as an entity which represents the basic unit of work to be implemented in the system.

Components of process are following.

|  |  |
| --- | --- |
|  |  |
| 1 | **Object Program** Code to be executed. |
| 2 | **Data** Data to be used for executing the program. |
| 3 | **Resources** While executing the program, it may require some resources. |
| 4 | **Status** Verifies the status of the process execution.A process can run to completion only when all requested resources have been allocated to the process. Two or more processes could be executing the same program, each using their own data and resources. |

**Process States**

As a process executes, it changes state. The state of a process is defined as the current activity of the process.

Process can have one of the following five states at a time.

|  |  |
| --- | --- |
|  |  |
| 1 | **New** The process is being created. |
| 2 | **Ready** The process is waiting to be assigned to a processor. Ready processes are waiting to have the processor allocated to them by the operating system so that they can run. |
| 3 | **Running** Process instructions are being executed (i.e. The process that is currently being executed). |
| 4 | **Waiting** The process is waiting for some event to occur (such as the completion of an I/O operation). |
| 5 | **Terminated** The process has finished execution. |

