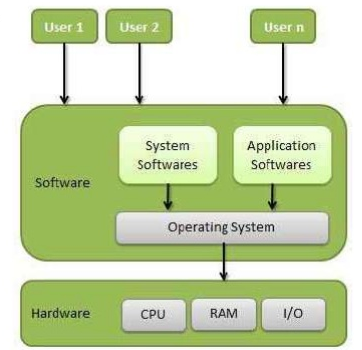
**Chapter1**

**Overview of operating system**

**What is operating system?**

An **operating system (OS)** is [system software](https://en.wikipedia.org/wiki/System_software) that manages [computer hardware](https://en.wikipedia.org/wiki/Computer_hardware) and [software](https://en.wikipedia.org/wiki/Computer_software) resources and provides common [services](https://en.wikipedia.org/wiki/Daemon_(computing)) for [computer programs](https://en.wikipedia.org/wiki/Computer_program).



**List any four functions of operating system.**

**1. Process Management** – Managing the programs that are running.

**2. Memory Management** – Managing and rationing the memory between processes and data.

**3. Storage Management** – Managing the permanent Storage of data on disks or other media

**4. I/O Management** – Managing the input and output

**5. Device / Resource Management** – Managing devices and resources and allowing the users to share the resources

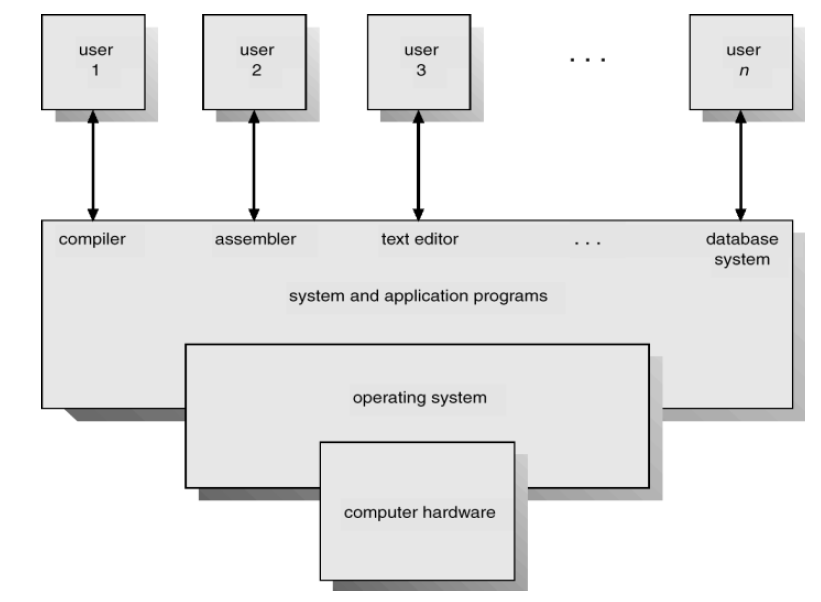
**6. Security and Protection** – Securing the system against possible unauthorized access to data or any other entity. Protecting the parts of the system against damage.

**7. Booting the System and getting it ready to work.**

**8. Data communications –** Providing interface to connect to other computers or allowing others to connect

|  |  |
| --- | --- |
| **List and draw a neat labelled diagram of four components of a computer system.** |  |

|  |
| --- |
| A computer system can be divided into four components:  1) The hardware.  2) The operating system  3) Application programs  4) The users |



**List different types of operating systems**

**Different types of operating systems:**

• Mainframe systems

• Multiprocessor systems

• Clustered systems

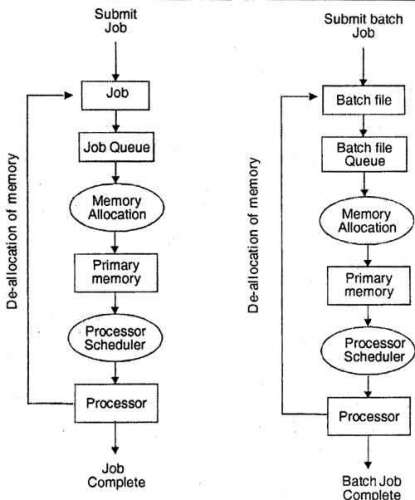
• Distributed systems

• Real time systems

**Explain batch operating system.**

A batch operating system normally reads a stream of separate jobs (from a card reader. For example), each with its own control cards that predefine to prevent errors and improper use of the computer. It is concerned with the operation and control if I/O devices.

* A batch system is one in which jobs are bundled together with the instruction necessary to allow them to be processed without intervention. Often jobs of a similar nature can be bundled together to further increase economy.
* Common input devices were card readers and tape drives. The basic physical layout of the memory of batch job computer is shown in fig.
* The OS was simple, its major task was to transfer control from one job to the next. The job was submitted to the computer operator in form of punch cards. At some later time the output appeared.
* The OS was always resident in memory. Often magnetic tapes and drums were used to store intermediate data and compiled programs. Example: Payroll system, stock control and billing systems.

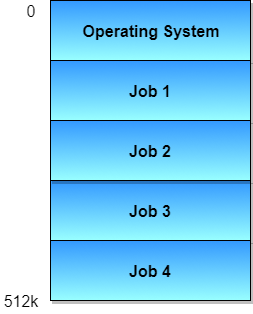


**Explain Multiprogramming operating system.**

* In this the operating system picks up and begins to execute one of the jobs from memory.
* Once this job needs an I/O operation operating system switches to another job (CPU and OS always busy).
* Jobs in the memory are always less than the number of jobs on disk (Job Pool).
* If several jobs are ready to run at the same time, then the system chooses which one to run through the process of CPU Scheduling.
* In Non-multiprogrammed system, there are moments when CPU sits idle and does not do any work.
* In Multiprogramming system, CPU will never be idle and keeps on processing.

Time Sharing Systems are very similar to Multiprogramming batch systems. In fact time sharing systems are an extension of multiprogramming systems.

In Time sharing systems the prime focus is on minimizing the response time, while in multiprogramming the prime focus is to maximize the CPU usage.



**Explain time sharing operating system.**

* A time sharing system allows many users to share the computer resources 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 rapidly from one user to the next, each user is given the impression that the entire computer system is dedicated to his use, even though it is being shared among many users.
* The time sharing systems were developed to provide an interactive use of the computer system.

**Example**:-



In above figure the user 5 is active but user 1, user 2, user 3, and user 4 are in waiting state whereas user 6 is in ready status. As soon as the time slice of user 5 is completed, the control moves on to the next ready user i.e. user 6. In this state user 2, user 3, user 4, and user 5 are in waiting state and user 1 is in ready state. The process continues in the same way and so on.

**Advantages of Time Sharing System**

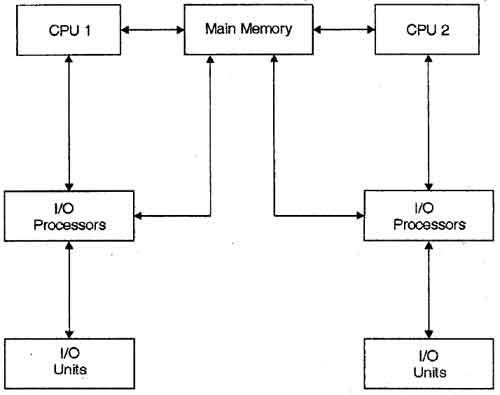
* Each user can get CPU time.
* Efficient CPU utilization.
* Time sharing systems were developed to provide interactive use of a computer system at a reasonable cost.
* A time shared operating system uses CPU scheduling and multi programming to provide each user with a small portion of a time-Shared Computer.

**Disadvantages of Time Sharing System:**

* The time-shared systems are more complex than the multi-programming systems.
* Context switching occurs frequently. i.e. Multiple processes are managed simultaneously which requires an adequate management of main memory so that the processes can be swapped in or swapped out within a short time.

**Explain Multiprocessor Operating System**

**Multiprocessor Operating System** refers to the use of two or more [central processing unit](http://ecomputernotes.com/fundamental/introduction-to-computer/what-is-cpu)s (CPU) within a single [computer](http://ecomputernotes.com/fundamental/introduction-to-computer/what-is-computer) system. These multiple CPUs are in a close communication sharing the computer bus, memory and other peripheral devices. The basic organization of multiprocessing system is shown in fig.

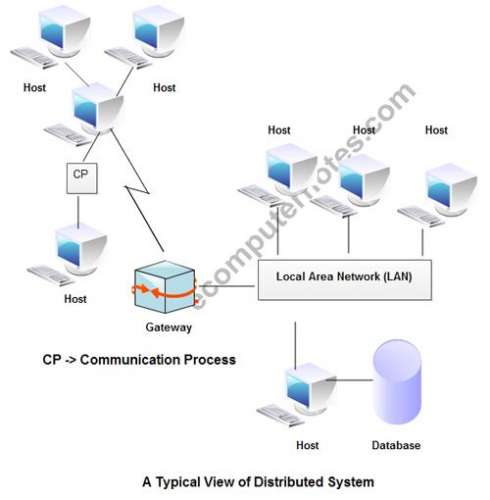


**Describe Distributed Operating system.**

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 –

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



The structure shown in fig contains a set of individual computer systems and workstations connected via communication systems, but by this structure we cannot say it is a distributed system because it is the software, not the hardware, that determines whether a system is distributed or not.

**Describe real time operating system in brief.**

Real time systems are used in environment where a large number of events, mostly external to the computer system, must be accepted and processes in a short time or within certain deadlines. Such applications include real-time simulations, flight control, industrial control, military applications etc.

A primary objective of real-time systems is to provide quick event response time and thus meet the scheduling deadlines. User convenience and resource utilization are of secondary concern to real time system designers. In Real time systems, processor is allocated to the highest priority process among those that are ready to execute. Higher priority processes preempt execution of the lower priority processes. This form is called as **‘priority –based preemptive scheduling’**.

**The primary functions of the real time operating system are to:**

**1.** Manage the processor and other system resources to meet the requirements of an application.

**2.** Synchronize with and respond to the system events.

**3.** Move the data efficiently among processes and to perform coordination among these processes.

**Types of real time system:**

**1. Hard real time**:-

Hard real time means strict about adherence to each task deadline. When an event occurs, it should be serviced within the predictable time at all times in a given hard real time system.

**Example: -**video transmission, each picture frame and audio must be transferred at fixed rate.

**2. Soft real time**:-

Soft real time means that only the precedence and sequence for the task operations are defined, interrupt latencies and context switching latencies are small. There can be few deviations between expected latencies of the tasks and observed time constraints and a few deadline misses are accepted.

**Example: - Mobile** phone, digital cameras and orchestra playing robots.

**Difference multiprogramming and multitasking**

|  |  |  |
| --- | --- | --- |
| **BASIS FOR COMPARISON** | **MULTIPROGRAMMING** | **MULTITASKING** |
| Basic | Enables multiple programs to utilize CPU simultaneously. | A supplementary form of the multiprogramming system also supports user interaction. |
| Objective | Improvement in CPU utilization. | Minimize the response time. |
| Switching | Occurs when the currently running process stops. | Carried out each time after the time slice of process is finished. |
| Complexity | Simple | Complex |

### **Difference between Multi-programming OS and Time sharing OS**

|  |  |
| --- | --- |
| **Multi-programming OS** | **Time sharing OS** |
| The process can be executed by a single processor | In this process two or more users can uses a processor on their own terminal |
| It has no fixed time slice | It has fixed time slice |
| In case of multi-programming before finishing a task the execution power is not taken off | The execution power is taken off before finishing of execution |
| Example: Mac OS | Example: Windows NT |

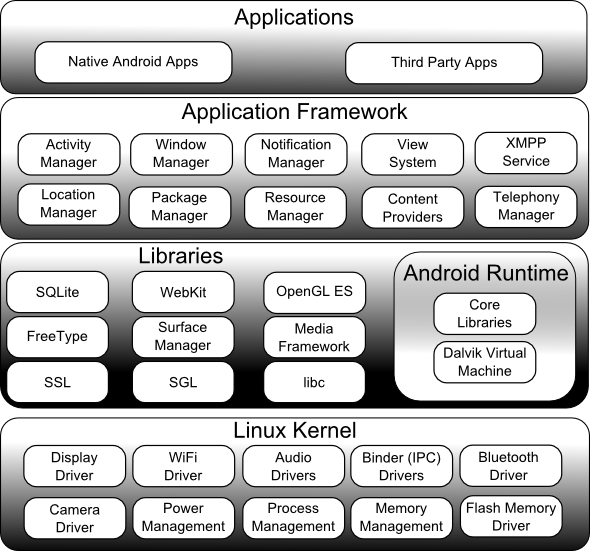
**Explain Mobile OS**

 A mobile operating system is the software platform on top of which other programs can run on mobile devices. The operating system is responsible for determining the functions and features available on your device

The Android Operating System is a Linux-based OS developed by the Open Handset Alliance (OHA).

The Android mobile operating system is Google's open and free software stack that includes an operating system, middleware and also key applications for use on mobile devices, including smartphones.

**Android OS Architecture**



Android operating system comprise of different software components arranges in stack. Different components of android operating system are –

1. Linux kernel
2. Libraries
3. Android Run time
4. Application Framework
5. Applications

**Linux kernel**

1. Bottom layer of android operating system is Linux kernel.
2. Android is built on top of Linux 2.6 Kernel and few architectural changes made by Google.
3. Linux Kernel provides the basic system functionality such as process management, memory management and device management
4. Linux kernel also provides array of device drivers which make our task easier while interfacing the android with peripheral devices.

**Libraries**

1. On the top of Linux Kernel another layer called libraries is present.
2. It provides the different libraries useful for well-functioning of android operating system.
3. Libraries are java libraries build specific for android operating system.

**Android Runtime**

1. It is third component of the android architecture and placed in second layer from bottom
2. It provides most important part of android called Dalvik Virtual Machine.
3. Dalvik Virtual Machine is similar to JVM but only difference is that it is designed and optimized for Android.
4. Dalvik Virtual machine uses core functions of Linux such as memory management and multithreading and enables each android app to run its own process.

**Application framework**

1. It is 2nd topmost component in android operating system stack
2. Android applications directly interacts with application framework.
3. Application framework manages the basic functions of android device such as resource management, voice call management etc.

**Applications**

Applications created by third party users or developer will be installed on application layer