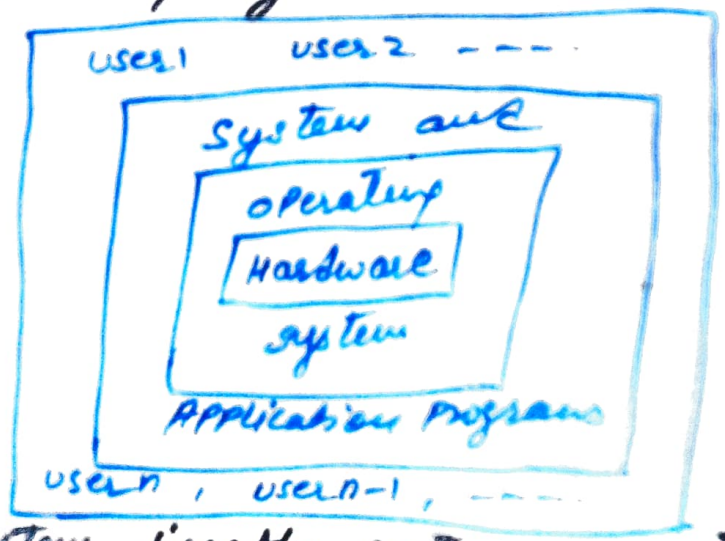
Operating System :- OS is a system s/w which manages all the activities of the computer. It is an interface between H/w & user in computer system.

It is a large collection of software which manages resources of the computer system such as memory, processor, file system and input/output devices. It keeps track of the status of each resource and decides who will have a control over computer resources for how long and when. OS works in four steps:

1) It keeps track of all resources.
2) It decide policy to allocate a resource to a process.
3) It allocate the resource to a process.
4) It deallocate the resources.

In short the operating system is one program running at all times on the computer (usually called the kernel) with all else being systems programs and application programs.



Operating system directly controls computer h/w resources

Basic Terms :-

Multiprogramming :- Multiprogramming is a technique to execute number of programs simultaneously by a single processor. In multiprogramming number of processes are reside in main memory at a time. The OS picks and begin to execute one of the jobs in the main memory. Whenever the job does not need CPU i.e. the job has to do only with I/O devices, and the CPU is idle at that time, the OS switches to another job in the memory and CPU executes a portion of it till this job issue a request for I/O or the first one has finished its I/O.

Multitasking :- is an logical extension of multiprogramming to perform multiple task at a same time or at different time. Multitasking is used when more than one applications / programs / tasks are being run on a single processor.

Multiuser :- is a term that defines an operating system or application software that allows concurrent access by multiple users can work on time or different time. In other words it allows two or more users to run programs at the same time.

Multiprocessing :- capable of supporting and utilizing more than one computer processor, with in a single computer. Multiple jobs by several CPUs or ability to use more than one processor on a single machine.

Multiprocessing → symmetric
Multiprocessing → Asymmetric

Asymmetric multiprocessing :- it is which each processor is assigned a specific task. A master processor controls the system; the other processors either look to the master for instruction or have predefined tasks. This schema defines a master slave relationship.

Symmetric Multiprocessing :- it is which each processor runs an identical copy of the operating system, and those copies communicate with one another as needed.

Multi Access :- Multiaccess operating systems allow simultaneous access to a computer system through two or more terminals. In general multiaccess operation does not necessarily imply multiprogramming.

EVOLUTION OF OPERATING SYSTEM :- An operating system may process its workload serially or concurrently.

(i) SERIAL PROCESSING :- Before 1950 the programmers directly interact with hardware, there is no operating system that time. If the programmer wish to execute a program on those days, the following serial steps are necessary.

⇒ Type the program on punched card.
⇒ Convert the punched card to card reader.
⇒ Submit to the computing machine, is there any errors, the error condition was indicated by lights.
⇒ The programmer examine the registers and main memory to identify the cause of the error
⇒ Take the output on the printers.

Then the programmer ready for next program.

Disadvantages :-

⇒ This type of processing is difficult for users.
⇒ It takes much time and next program should wait for completion of previous one.

(ii) BATCH PROCESSING :- By reducing or eliminating component idle times due to slow manual operations, batch processing

offers a greater potential for increased system resource utilization and throughput than simple serial processing, especially in computer systems that serve multiple users.

⇒ Operating system commands are statements written in Job control language (JCL). Typical JCL commands include marking of job beginning and end, commands for loading and execution of programs, and commands to announce resources needs such as expected execution time and memory requirement

⇒ A memory resident portion of the batch OS - sometimes called the batch monitor - reads, interprets and executes these commands.

⇒ In response to them, batch jobs are executed one at a time.

⇒ A job may consist of several steps, each of which usually involves - loading and execution of a program.

MULTI PROGRAMMING :- Multiprogramming is used for optional utilization of CPU time and input output devices. A multiprogramming environment allows the idle time between the execution of a program to be used by the another program. In non-multiprogramming system, the CPU can execute

only one program at a time, if the running program at a time, if the running program waiting for any I/O device, the CPU become idle, so it will effect on the performance of CPU. But in multiprogramming it executes no. of programs simultaneously by a single processor.

Advantages of Multiprogramming :-

- Can get efficient memory utilisation.
- CPU is never idle, so the performance of CPU will increase.
- Throughput of the CPU may also increase.
- In non-multiprogramming environment the user program has to wait for CPU much time. But waiting time is limited in multiprogramming.

TIME SHARING SYSTEMS :- Time-sharing is a logical extension of multiprogramming & multiuser system. The CPU execute multiple jobs by switching among them, but the switches occur so frequently that the user can interact with each program while it is running. A time shared operating system allow many users to share the computer simultaneously multitasking is also possible

for a system. In this method the CPU time was shared by different processes. So it is said to "Time sharing Systems".

Advantages :-

- ✓ User can interact with the job when it is executing, but it is not possible in batch System.
- ✓ Efficient CPU utilisation
- ✓ To provide interactive use of computer system at a resonable cost.

PARALLEL SYSTEMS :- There is a trend towards multiprocessor systems. Such systems have more than one processor in close communication, sharing the computer bus, the clock and some times memory and peripheral devices. These systems are referred as tightly coupled systems. A system consisting of more than one processor and it is a tightly coupled then the system is parallel system. There are several reasons for buildings such systems. One advantage is increased "throughput". The number of jobs are completed by a CPU within a time period is said to be throughput. Another advantage for multiprocessor is that they increase reliability.

DISTRIBUTED SYSTEMS :- In distributed systems, the processors can't share memory or a clock, each processor has its own local memory. The processor communicate with one another through various communication lines. Such as high speed buses. These systems are usually referred to as loosely coupled system or distributed system.

Advantages of distributed System :-

- Resource sharing
- computation speed up
- Increased Reliability
- Communication

REAL TIME SYSTEMS :- Real time operating system are used in environments where a large number of events, mostly external to the computer system, must be accepted and processed in a short time or within certain deadlines. A primary objective of real time systems is to provide quick event-response times, and thus meet the scheduling deadlines. User convenience and resource utilization are of secondary concern to real time system designers. "Real time OS has well defined fixed time limit, i.e. processing must be done with in the defined time limit otherwise the system will fail."

To understand more fully the operating System's role, we explore OS from two viewpoints

- User View
- System View

USER VIEW :- The goal of operating system from the user point of view is to maximize the work (or play) that the user is performing. So, the OS is designed mostly for ease of use, with some attention paid to performance and none paid to resource utilization.

A user sits at a mainframe or mini-computer. These users share resources and may exchange information. The OS is such cases is designed to maximize resource utilization - to assure that all available CPU time, Memory and I/O are used efficiently and that no individual user takes more than her fair share.

SYSTEM VIEW :- From the computer's point of view, the OS is the program most intimately involved with the hardware. In this content, an OS as a resource allocator. A computer system has many resources that may be required to solve a problem : CPU time, memory space, file-storage space, I/O devices and soon. The OS acts as the manager

of these resources.

A slightly different view of an operating system emphasizes the need to control the various I/O devices and user programs. An OS is a control program. A control program manages the execution of user programs to prevent errors and improper use of the computer. It is especially concerned with the operation and control of I/O devices.

OS WORKS ON 2 MODES → user mode
→ kernel mode (Supervisor/System/privileged)

mode bit :- A bit called the mode bit, is added to the hardware of the computer to indicate the current mode : kernel (0) or user (1).

✓ When the computer system is executing on behalf of user application, the system is in user mode. However, when a user application requests a service from the operating system it must transition from user to kernel mode to fulfill the request.

✓ At the boot time, the hardware starts in kernel mode. The OS is then loaded and starts user applications in user mode.

✓ Whenever a trap or interrupt occurs, the h/w switches from user mode to kernel mode.

FUNCTIONS OF OS :- The operating system provides an user friendly environment for the creation and execution of programs and provide services to the user. The main functions of operating systems are :-

(i) Program creation :- The OS provides editors, debuggers to assist the programmer in creating programs.

(ii) Programs execution :- A number of tasks required to execute a program, the tasks include instructions and data must be loaded into main memory, I/o devices and files must be initialized and other resources must be prepared.

(iii) Input / output operations :- A running program may require input output. This I/o may involve a file or an I/o device. A user program cannot execute I/O operations directly, the OS must provide some means to do so.

(iv) Error detection :- The operating system detects the different types of errors and should take appropriate action. The errors include memory error, power failure, illegal instruction in the program etc

(v) Resource allocation :- The OS collects all the resources in the network environment or a system and these resources to requested process.

(vi) **Accounting** :- The operating system can keep track of which users can how much and what kind of computer resources.

(vii) **Protection** :- The OS provides security Mechanism to protect the unauthorized usage of files in the network environment.

x - x - x - x - x - x - x - x - x - x - x

OPERATING SYSTEM SERVICES :- Operating System services are divided into two broad classes :-

→ System call

→ System Program

SYSTEM CALL :- System call is an interface between running process and OS. If a process needs a service from OS it issues a system call. System calls allow user-level processes to request some services from the OS which process itself is not allowed to do.

USE OF SYSTEM CALL :-

How System calls are used for copy a file

Example System Call Sequence

```
Acquire input file name
    write prompt to screen
    Accept input
Acquire output file name
    write prompt to screen
    Accept input
open the input file
    if file doesn't exist, abort
create output file
    if file exists, abort
Loop
    Read from input file
    write to output file
Until read fails
close output file
write completion message to screen
Terminate normally
```


API :- Systems and applications programmers often invoke services of the operating system from their programs by means of system calls, which are sometime called Application Programming interfaces (APIs). The functions that make up an API typically invoke the actual system calls on behalf of "App" programmer

Why an application programmer prefer to call an API rather than invoking actual System call?

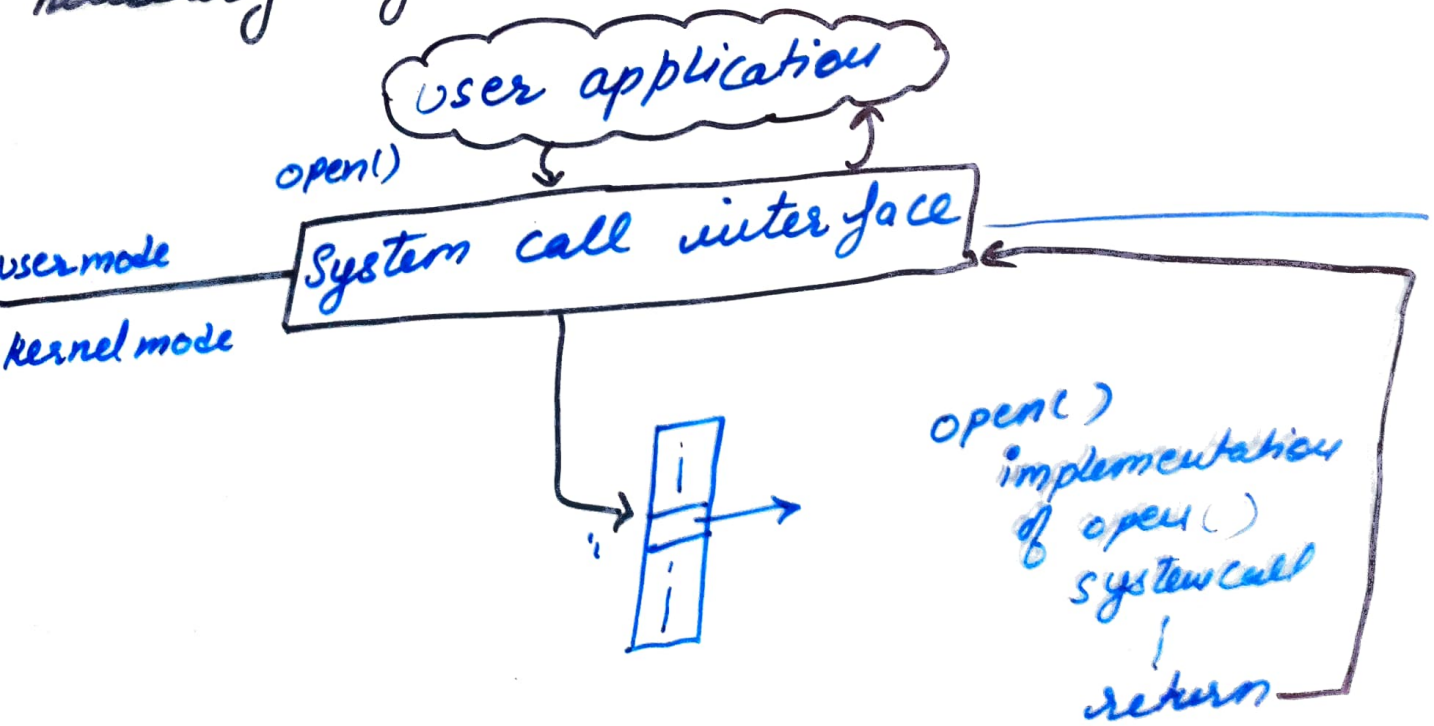
Ans Portability :- An application programmer expect her program to compile & run on any system that supports the same API.

Difficult :- Actual system calls can often be more detailed and difficult to work with the API available to an application programmer.

x - x - x - x - x - x - x - x - x - x

The run time supports system for most programming languages provides a system call interface that serves as the link to system calls made available by the OS.

The system call interface intercepts function calls in the API and invoke the necessary system call within the OS.



Note :- System calls occur in different ways depending on the computer in use.

Three general methods are used to pass parameters to the operating system.

→ Pass the parameters in registers.
→ Address of the block is passed as a parameter in a register.
→ Parameters also can be placed or pushed onto the stack by the program and popped off the stack by the OS.

TYPES OF SYSTEM CALLS :- System calls can be grouped into five major categories :-

(i) Process Control :- create, terminate, load, execute, end, abort, allocate & free memory, wait, signal.

(ii) File Management :- Create, delete, open, close, read, write in a file.

(iii) Device Management :- read, write, request, release

(iv) Information Maintenance :- date, time, get data, set data, get process, file, set process, file or device attributes.

(v) Communication :- create, delete, send, receive, transfer status, attach or detach remote device.

SYSTEM PROGRAM :- System programs provide a convenient environment for program development and execution. System programs provide basic functioning to users so that they do not need to write their own environment for program development (editors, compilers) and program execution (shells). In some sense they are bundles of useful system calls. They can be divides into these categories:

File Management :- These programs create, delete, copy, rename, print dump, list and generally manipulate files and directories.

Status Information :- date, time, disk space, no. of users, status information, logging & debugging information. Some systems also support a registry which is used to store and retriere configuration information.

File modification :- several text editors may be available to create and modify the content of files stored on disk or other storage devices.

Programming Language support :- Compilers, assemblers, debuggers and interpreters for common programming languages.

Program loading & execution :- System may provide absolute loaders, relocatable loaders, linkage editors and overlay loaders.

Communications :- These programs provide the mechanism for creating virtual connections among processes, users and computer systems.

System calls	System Programs
(1) System calls allow user level processes to request some services from the OS which process itself is not allowed to do.	(1) System programs provide basic functioning to users so that they do not need to write their own environment for program development & program execution.
(2) A System call is a request to the OS to do something.	(2) A System Program is a program that provides services to other programs.
(3) A system call is a fun^{n} that is being invoked in the running system code.	(3) A system Program is some where in the system storage and not necessarily running.
(4) System call will cause CPU running level switch, from user level to kernel level.	(4) System program maybe is simple library function like sins, cons.