What is an operating System?

It is a collection of software that directs a computer's operation controlling & scheduling the execution of other programs & managing storage, i/p and o/p & communication resources.

Objectives of OS

A program that controls the execution of application programs.

An interface blw applications & hlw.

The following objectives of OS:

1) Convenience: Makes the computer more convinient to use.

2) Efficiency :- Allows computer system resources to be used in an efficient manner.

3) Ability to evolve:- permit effective development, testing & introduction of new system functions without interfering with service.

Functions of O.S

1) Service providing :- provide standard facilities like system, standard libraries, window system.

2) coordinator :- Three aspects.

protection:- protect oops from interfering with each other.

Communication: enable jobs to interact with each other.

Resource management :- facilitate sharing of resources across jobs.

3) OS are everywhere:-

Single function devices (embedded controllers),

- OS provides a collection of standard services
- sometimes OS/middleware distinction is blurry.
- Multifunction/application device (Work stations & servers)
- OS manages application interactions.

Why do we need OS?

→ convinience

→ Efficiency

→ Booting a computer.

OS:- It is a program that manages the computer H/w.

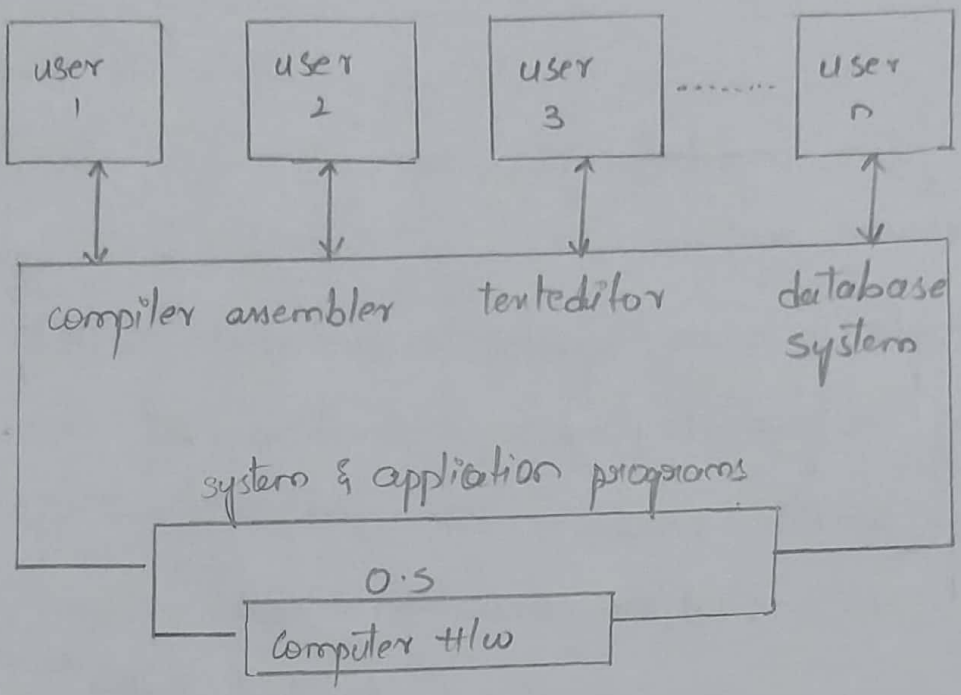
→ It acts as intermediatory b/w the computer user & computer H/w.

→ OS is designed to provide an environment in which user can easily interface with computer to execute prgms.

→ O.S is designed to be 1) convinient 2) efficient.

→ computer system can be roughly divided into 4 components

1) H/w
2) O/S
3) Application programs
4) users.



→ Hlw means CPU memory, i/p & o/p devices

Why we use O.S?

→ The goal of computer system is to execute user programs & to make OS solving user problems easier.

→ These pgms require certain common operations such as controlling Ilo devices.

→ The common functions of controlling & allocating resources are brought together into one piece of slw the OS.

→ one pgm running all the times on the computer is called Kernel.

→ along with Kernel 2 types of programs

1) Sys pgms: Which are associated with os & but are not part with Kernel.

2) Appn pgms:- When include all programs not associated with O.S.

Computer System architecture:-

1) single-processor systems:

→ In the single processor systems is one main CPU is capable of executing a general purpose instruction set.

→ The device-specific processors such as disk, keyboard & graphics controllers on mainframes.

→ Special-purpose processors run a limited instruction set.

→ These are managed by operating system.

→ Special-purpose processors are low level components built in the hardware.

2) Multiprocessor Systems:-

Multiprocessor systems also known as parallel systems or tightly coupled systems.

→ Two or more processors in close communication, sharing the computer bus, clock, memory & peripheral devices.

Multiprocessor systems have three main advantages.

1. Increased through put:-

Increasing the number of processors & get more work done in less time.

2) Economy of scale:-

Its cost is less than multiple single-processor system because they share peripherals, mass storage & power supplies.

3) Increased reliability:-

The functions can be distributed properly among several processors, then the failure of one processor will not halt the system.

→ They multiprocessors are of two types

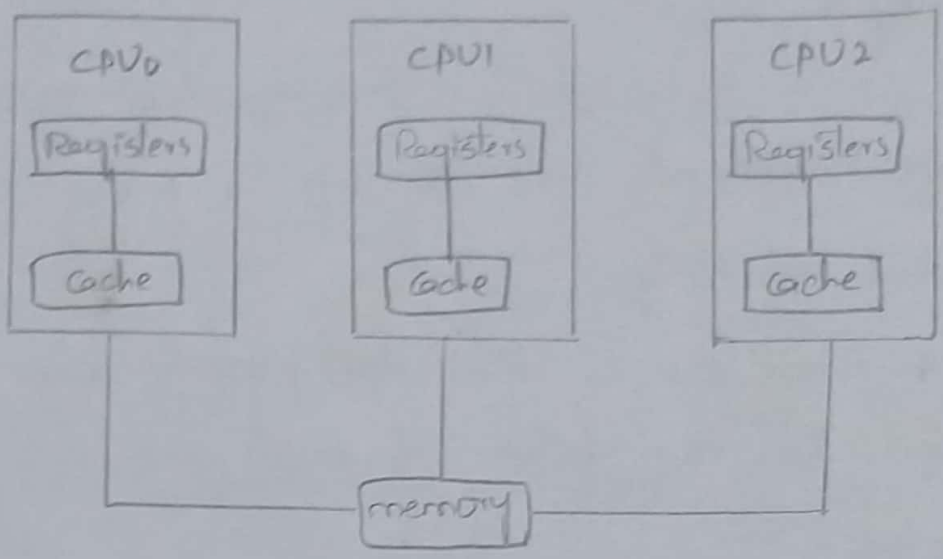
1) Asymmetric multiprocessing.
2) Symmetric multiprocessing.

1) Asymmetric multiprocessing:-

→ Each processor is assigned a specific tasks.
→ A master processor control the system.
→ The other processor look to the master for instruction.
→ The masters processor schedules & allocate work to be slave processors.
→ This scheme is called master-slave-relationship.

2) Symmetric multiprocessing (SMP):-

Each processor performs all tasks within the operating systems.
→ SMP means that all processors are peers. no master-slave relation-ship exists blw processors.



Symmetric multiprocessing architecture

3) Clustered systems

→ clustered system is one of the multiple-CPU system.

→ clustered computers share storage through LAN.

→ It is used to provide high-availability service. This service will continue if one of or more systems in the cluster fail.

→ clustering is divided into ① assymmetric ② symmetric

① Assymmetric clustering:- one machine is in hot-standby mode while the other is running the applications.

→ This hot-standby host machine does nothing but monitor the active server.

→ If the server fails, the hot-standby host becomes the active server.

2) Symmetric clustering:-

→ Two or more tasks are running applications & monitoring each other.

→ It is more efficient.

→ It require more than one application to run.

Operating system structure:-

→ multiprogramming :-

→ Multiprogramming increases CPU utilization by organizing Jobs so that the CPU always has one to execute.

→ The operating system keeps several Jobs in memory.

→ The main memory is the disk is too small to accomodate all Jobs, the Jobs are kept initially on the disk in the Job pool.

→ This pool consists of all processes residing on disk awaiting allocation of main memory.

→ The set of Jobs in memory can be a subset of the Jobs kept in the Job pool.

→ The OS picks & begins to execute one of the Job in memory.

→ The Job may have to wait for some tasks, such as an I/O operation to complete.

→ In multiprogrammed system the OS simply switches to & execute another Job.

→ When that Job needs to wait, the CPU is switched to & another Job.

→ The first Job finishes waiting & gets the CPU back.

→ At least one Job needs to execute the CPU is never idle.

Time sharing (Multi tasking): It is a logical extension of multiprogramming.

→ In this CPU executes multiple jobs by switching among them these switches occur so frequently that the users can interact with each program while it is running.

→ It requires an interactive computer system which provides direct communication between the user & the system.

→ A time-shared operating system allows many users to share the computer.

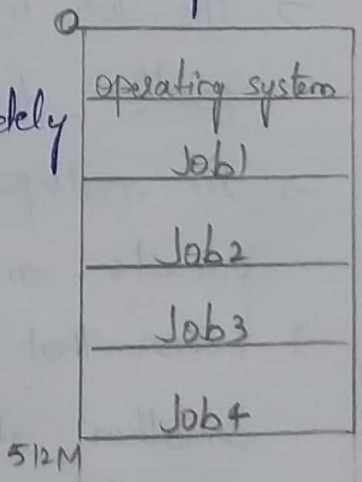
→ A pogm loaded into memory & executing is called process.

→ The operating system selects a job from the job pool it loads that job into memory for execution is called job scheduling.

→ Several jobs are ready to run at the same time the system must choose among them.

→ The processes are swapped in & out of main memory to the disk is called swapping.

→ The execution of a process that is not completely in memory is called virtual memory.



Memory layout for a multiprogramming system

Operating System Operations:-

1) Dual-mode operation:-

→ They are two modes:

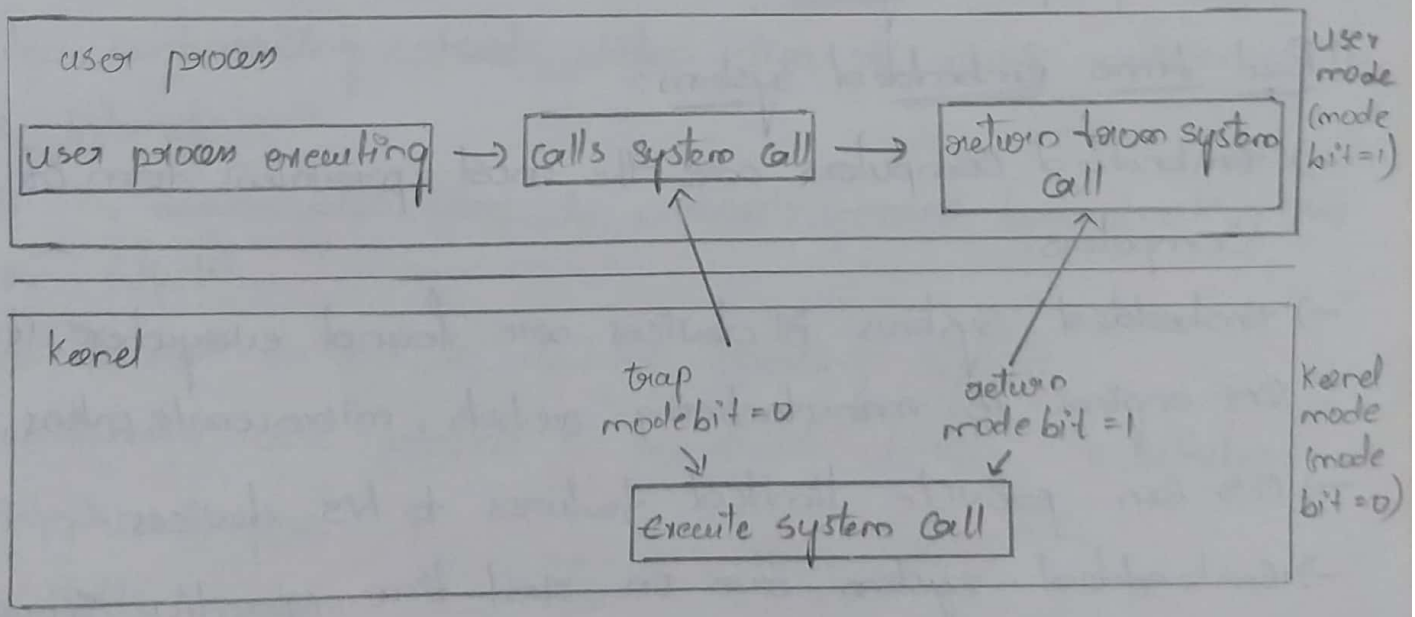
1) user mode 2) Kernal mode.

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

→ With the mode bit a tasks is executed on behalf of the operating system & one is executed on behalf of the user.

→ The computer system is executing on behalf of a user application then the system is in user mode.

→ When a user application requests a service from the operating system it transition from user to kernel mode to fulfill the request.



Transition from user to kernal mode.

Timer:-

→ Timer to prevent a user program from running too long.

→ A simple technique is to initialize a counter with the amount of time that program is allowed to run.

Eg: A prgm with a 7min time limit. counter initialized to 420 every second, time interrupt & the counter is decremented by 1.

→ The counter is positive, the control is returned to the user prgm.

→ The counter is negative the operating system terminates the program for exceeding the assigned time limit.

Special purpose Systems:-

1) Real time embedded systems:-

→ Embedded computers are the most prevalent form of computers.

→ Embedded systems of devices are found everywhere, ie., car engines to manufacturing robots, microwave ovens.

→ O.S can provide limited features to his devices.

→ Embedded system run on real-time operating services because they are used as control devices.

→ Sensors bring the data to the computes. The computes analyse the data & controls to modify the sensor inputs.

→ This system control scientific experiments, medical,

imaging systems industrial control systems and certain display systems are real-time systems.

→ A real time system has well-defined, fixed time constraints.
→ A real time system functions correctly only if it returns the correct result within the time constraints.

2) Multimedia systems:-

→ Most operating systems are designed to handle conventional data such as text files, programs, word-processing documents & spreadsheets
→ A recent technology is multimedia data into computes systems
→ Multimedia data consist of audio & video files as well as conventional files.

Applications:- It includes audio files such as MP3 DVD movies, video conferencing, short video clips of movie previews over the Internet.
→ It is also used in webcasts means broadcasting over the WWW.
→ It is not limited to desktop operating system
→ Multimedia also popular in handheld devices like cellular telephones, PDA

Handhelds systems:-

→ It refers to small portable devices, It can be carried & capable of performing normal operations, they are battery powered.

Eg:- personal digital Assistants (PDAs), mobile phones, palm-top Computers.

→ Handled devices is having certain limitation for weight and size as a result they are small memories, slow processor and small display screen etc.

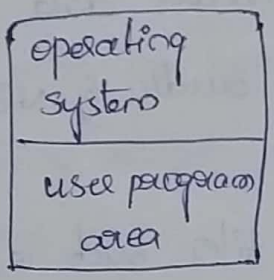
→ present days we are using wireless technologies like bluetooth, users are able to access the internet, email etc.

Evaluation of operating systems :-

i) simple Batch systems :-

→ The os in early computers was fairly simple. The major task was to transfer automatically from one job to the next.

→ The os always reside in memory.



memory layout for a simple batch system.

→ To speed up processing jobs with similar needs were batched together & run through the computer as a group.

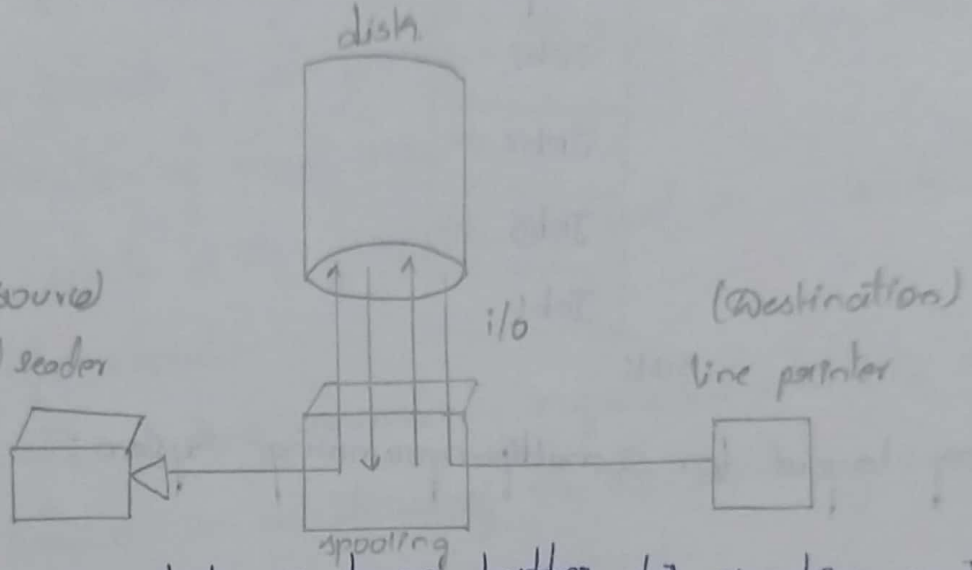
→ programmers leave their programs with the operator. The operator would sort programs into batches with similar requirements & computer became available to run each batch.

→ The definite feature of a batch system is the lack of interaction blw user & the job while the job is executing.

→ Job being processed cards are read directly from card reader onto the disk.

→ When the job requests the printer to output a line, that line is copied into a system buffer & is written to the disk,

→ When a job is completed the o/p is printed. This processing is called spooling.



→ spooling uses the disk as huge buffer for reading on i/p devices & for storing output files until the output devices are able to accept them.

→ It is used for processing data at remote sites.

→ In simple system the spooler is reading the input of one job & printing the output of a different job.

Multiprogrammed Batched Systems:-

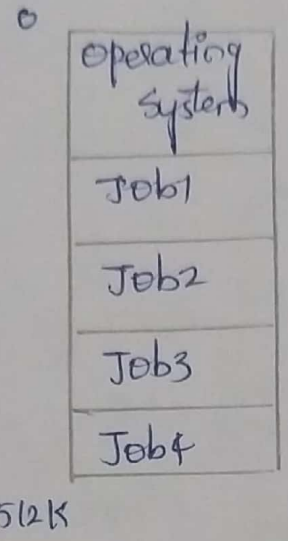
→ When jobs come in directly on cards or on magnetic tape it is not possible to run different order.

→ Jobs must be run sequentially first come first serve basis.

→ When several jobs are on a direct-access device such as a disk job scheduling. Job scheduling is the ability of multiprogram.

→ Multiprograming increases cpu utilization by organizing jobs such that cpu always has one to execute.

→ The OS keeps one always several jobs in memory at a time.



Memory layout for a multiprogramming System.

→ In this multiprogramming System, The OS simply switches to add executes another Job.
→ All the Jobs Enter into the system are kept in the Job pool.
→ This pool consists of all processes residing on mass storage waiting allocation of main memory.
→ when OS selects a Job from the Job pool it it loads that Job into memory for execution.
→ Multiple Jobs running concurrently require the ability to affect one another be limited in all phases of the OS. including process scheduling, disk storage & memory management.

4) Time-Sharing Systems :-
→ Time-Sharing or multi tasking is a logical extention of multi-programming
→ It was developed to provide interactive use of a computer system at a reasonable cost.
→ Time sharing as uses CPU scheduling & multiprogramming.
→ Multiple Jobs are executed by the CPU. It switches occur so frequently the user may interact with Each program while it is running.
→ Time sharing OS are more complex than the multi-programmed OS.

5) Personal-Computer Systems :-

→ A computer system dedicated to a single user. These types of computer system called as personal computer (PC's)

→ PC's are appeared in 1970's. PC's OS is neither multiuser nor multitasking.

→ OS goals are changed with this PC's by using maximize user convinence & responsiveness instead of CPU.

→ Eg:- MS-DOS operating system from Microsoft has superseded by multiple flowers of Microsoft Windows

6) Parallel Systems :-

→ A system have more than one processor in close communication sharing the computer bus, the clock, memory & peripheral devices. These systems are referred as tightly coupled systems

→ By using parallel systems increased throughout

→ Multiprocessor systems is that they increase reliability.

→ They are two types of multiple-processor.

① → Symmetric multiprocessing :- Each processor runs an identical copy of operating system & these copies communicate with one another as needed

② → Asymmetric multiprocessing :- Each processor is assigned a specific task master processor controls the system. The master processor schedules & allocates work to the slave processors.

7) Distributed Systems :-

→ Each processor has its own local Energy

→ The processor communicate with one another through various communicate lines such as high-speed bus or telephone lines. These system are referred as loosely coupled system or distributed system.

Resource Energy :-

It is distributed system provides mechanism for sharing files at remote sites, processing information in a distributed database Printing files at remote sites using remote specialized devices & performing other operations.

Computation speedup :-

→ Distributed system may allow us to distribute the computation among the various sites to run that computation concurrently.

→ If a particular site is currently overloaded with jobs, some of them moved to other lightly loaded sites. This movement of jobs is called load sharing.

Reliability :-

→ If one site fails in a distributed system, the remaining sites can potentially continue operating

Communication :-

→ There are many instances in which programs need to exchange data with one another on one system.

→ Eg :- Windows system frequently share data to transfer data b/w displays user many initiate file transfer or communication with one another via e-mail.

8) Real-Time Systems :-

→ Spacial purpose operating system is the real-time system.

→ It is used where there is rigid time requirements on the operation of a processor ~~or flow of data~~.

→ Sensors bring data to the computer.

→ System that control scientific experiments, medical imaging systems, industrial control systems & some display systems are real-time systems.

→ Processing must be done within the defined constraints or system will fail.

→ They are two of real-time systems.

① → Hard real-time systems :- It guarantees that critical task is complete on time.

→ Virtual Memory never found on real-time systems.

→ If conflict with the operation of time-sharing systems & the two cannot be mixed.

② → Soft real-time System :- where a critical real-time task's get priority over other tasks & retains that priority until it completes.

→ The Unix os which is written mostly in 'c' the other hand is available on a number of different CPU's including Intel 80x86 Motorla 680x0//

Operating System Services :-

→ It provides an Environment for the execution of program
→ The specific services different from one operating system to another
→ The os services are provided for the Convenient of the programmer to make programming task easier

1) User Interface :- All operating have a user interface.
→ Interface can take several forms one is a command line interface which uses text commands.
→ Batch interface in which commands & directives to control those commands are entered into files
→ Most commonly used is Graphical user Interface (GUI)
→ Here the interface is a window system with a pointing device to direct-I/O, choose from menus & makes selections & a keyboard to Enter text.

2) **Program Execution** :- The system must be able to load a program into memory & to run the program. The program must be able to End its execution normally or abnormally.

3) **I/o operations** :- I/o means any file or any specific I/o device program may require any I/o device while running

—> so, os must provide the required I/o. //

- File System manipulation :- program needs to read a file or write a file The os gives the permission to the prgm for operation on file.

Communication :- Data transfer b/w two processes is required for some time. Communication may be Implemented by two method shared memory & message passing

Error detection :- Error may occur in CPU, in I/O devices or in the memory hardware in I/O devices & in the user program

→ For each type of error the os should take the appropriate action to ensure current & consistent computing.

Resource allocation :- when these are multiple user's are running at the same time, resources must be allocated to each of them

→ Many different types of resources are managed by the OS.

Accounting :- we want to keep track of which users use how much and what kinds of computer resources.

→ This record keeping may be used for accounting.

Protection & Security :- The owner of information stored in a multiuser or n/w computer system

→ when several separate processes execute concurrently, it should not be possible for one process to interfere with the other or with os itself.

→ Protection involves ensuring that all acess to system resource is allocated

→ Security of the system from outsiders is also important.

System calls :- System calls are defined as mediator or interface b/w a process & a O.S.

→ Initially system calls were written in assembly languages but now they are written in high level language like C, C++ etc

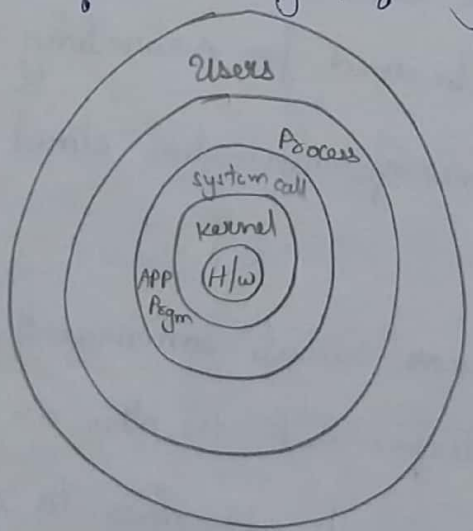
→ All these system calls are present in a library known as API i.e, Application programme interface.

→ Examples are how a system call works.

First example :- Let us assume that user has written a prgrm in 'c' to read a file, when this program is executed, the read function call makes a call to read a function present in the c library.

→ This read call involves a switch from user mode to kernel mode because of generation of system call.

→ The read call of windows API is executed & results are written back to read call of c Library & finally to user.



· Second Example :- Let us consider second example that if user wants to copy data from one file i.e source file to other file i.e, destination file

→ The user has to read data from one file & copy data to other file.

→ First it finds names of two files

→ so the os has to give a message on monitor using write system call.

→ when user sees the message on monitor enter the name of the files which is accepted by the os by using read system call.

→ when user sees the message on monitor enter the name of the files which is accepted by the os by using read system call

→ one the names of two files are obtained then the os opens the input file by using opens() system calls, & creates an output file by using creates() system call, if i/p file exists.

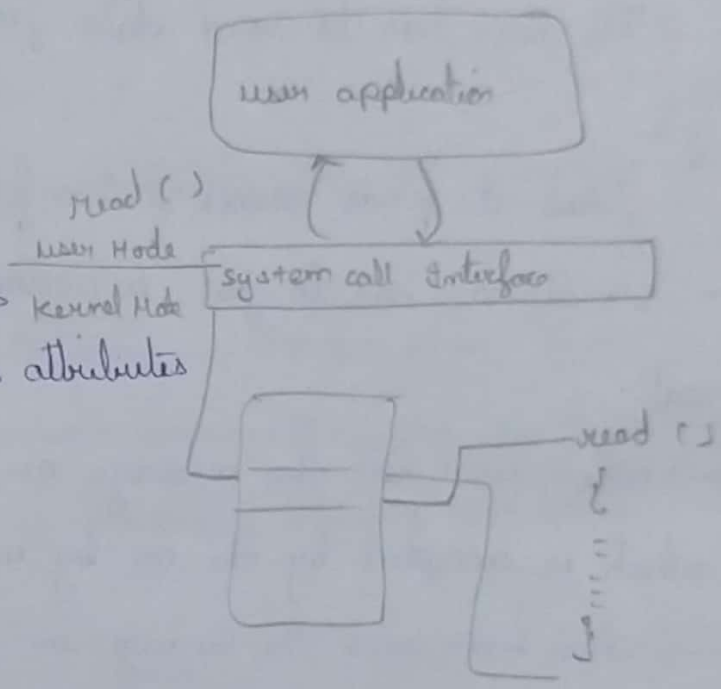
→ otherwise a system call is generated & prgrm terminates by printing a message on monitor that file does not exists

→ If an o/p file with same name is present then os involves a a system call & asks user whether to overwrite that file, If response is yes then file is overwritten.

→ Both the files are present now read() operation involves read system call, that read data line by line to another file

→ The read gets completed once end of file is reached. The writing is done with help of write system call

Types of System calls :-



1) Process Control :-
→ End, abort
→ load, execute
→ create process, terminate process
→ get process attributes, set process attributes
→ wait for time
→ wait event, signal event
→ allocate and free memory

2) File management :-
→ Create file, delete file
→ open close
→ read, write, reposition
→ get device attributes, set device attributes

3) Device Management
→ request device, release device
→ read, write, reposition
→ get device att, set device att
→ logically attach or detach devices

4) Information maintenence :-
→ get time or data, set time or data
→ get system data, set system data
→ get process file, or device attribute → Set process, file or device attributes

Communication :-

→ Create, delete communication Connection

→ Send, recieve messages

→ transfer status, information

→ attach or detach remote devices

Process Control :- A running program needs to halt its execution either normally (end) or abnormally (abort)

→ If a system call is made to terminate the currently running program abnormally or if the program runs into a problem & causes an error trap a dump of memory is sometimes taken & an error msg generated.

→ The dump is written to disk & may be examined by a debugger It is a system program designed to aid the programmer in finding & correction bugs.

→ Load & Execute system calls are used by process to execute one program. This feature allows the command interpreter to execute a program.

→ Get process attributes & set process attributes system calls are used to determine & reset the attributes of a process It also includes process priority its maximum allowable execution time etc.,

→ wait time & wait event are the system call used in waiting condition of the process. when process is created it may need to wait for them to finish their execution.

→ Certain amount of time is required to wait before to complete the operation.

→ MS-DOS OS is an example of a single tasking system, which has a command Interpreter that is invoked when the computer is started.

→ To run a program MS-DOS uses simple method.

→ It does not create a new process while running one process

→ Unix is an example of multitasking system.

→ In Unix the command interpreter may continue running while another program is executed.

→ Fork system call is used to create a new program.

File Management :- System calls used for create & delete the file system calls require name of the file with file attributes for creating & deleting files.

→ other operations on files is read a file, write & reposition the file after it open.

→ For directories, same set of operations are performed sometimes we require to reset some of the attributes on files & directory

→ The system call get file attribute & get file attribute

Device Management :- For accessing device, system calls are used.

→ Many of the system calls for files are also needed for devices

→ In multiuser environment, request is made before to use of that device

→ After using any device it must be release.
So, using release system call device is free to use by another user
→ Read, write & reposition system calls may be used with device

Information Maintenance :- Information maintence system call includes calls to get/set the time, date, system date & process, files are device attributes

→ The os keeps information about all processes & system calls that are used to access this information

Communication :-

The message passing model must support calls to
→ Identify a remote process with which to communicate
→ Establish a connection b/w the two processes
→ open & close the connection as needed
→ wait for incoming messages in either a blocking or non-blocking
→ Delete for connection when no longer needed. the shared Memory.
→ Create & access memory that is shared amongst processor
→ Message passing is simples & easier.
shared memory is faster & & where large amount of data are to be shared.

System Programs :-

Modern os consists of a collection of system programs.

→ Some of them are simply user Interfaces to system calls.

→ It provides a convenient environment for program development & execution.

1) File Management : It programs are used to create, delete, copy, rename, list dump on files and directory.

2) Status Information :- By using this system program covers be date, time, disk space available memory & users.

→ All these information is formatted & display on output device.

3) File Modification :- Text Editors are used for file modification. In this new file is created & content of file is modified

4) Programming language support :- it includes the compilers, assemblers, interpreter for common programming language like C, C++, Java.

5) Programming Loading & Execution :- Program is assembled or compiled it must be loaded into memory to be executed.

→ The system may provide absolute loaders, relocatable loaders, linkage editors.

→ Debugging systems for either higher level languages or machine language are also needed

6) Communication :- It provides the mechanism for creating virtual connection among process, users & different computer systems

→ These programs allows users to send message to browse web page & to send email messages

→ From os point of view, system program is a command interpreter. It is used to get & execute the user specified command

System Design and Implementation :-

→ The problems of designing & Implementing a system. No complete solutions to the design problems but there are approaches that have been successful.

Design goals :-

→ The first problem in designing a system is to define the goals & specification of the system. The design of the system will be affected by the choice of hardware & type of system ; batch, time-shared, single-user, multi user, distributed, real-time or general purpose.

→ The highest design level the requirement may be much harder to specify. The requirements can be divided into two basic groups

1) User goals & system goals

→ User desire certain properties in a system. The system should be convenient to use, easy to learn, easy to use, reliable, safe & fast. These specifications are not particularly useful in the system design.

→ A similar set of requirements can be defined by those people who must design, create, maintain & operate the system

→ The os should be easy to design, implement & maintain. it should be flexible, reliable, error free & efficient

Implementation :-

→ once an operating system is designed it must be implemented

→ os have been written in assembly language. os can now be written in higher-level languages

→ The advantages of using a higher-level language. For implementing operating system the language is used for application programs.

→ The code can be written faster and is easier to understand & debug.

→ The compiler technology will improve the generate code for entire os by simple recompilation.

→ Finally os is for easier to port-to move to some other H/w if it is written in high-level languages

→ Eg :- MS-DOS was written in Intel 8088 assembly language it is available only Intel family of CPU's.

→ If sufficient redundancy exist in the system, i.e both hardware & data the system can continue its operation even if some of its sites have failed.

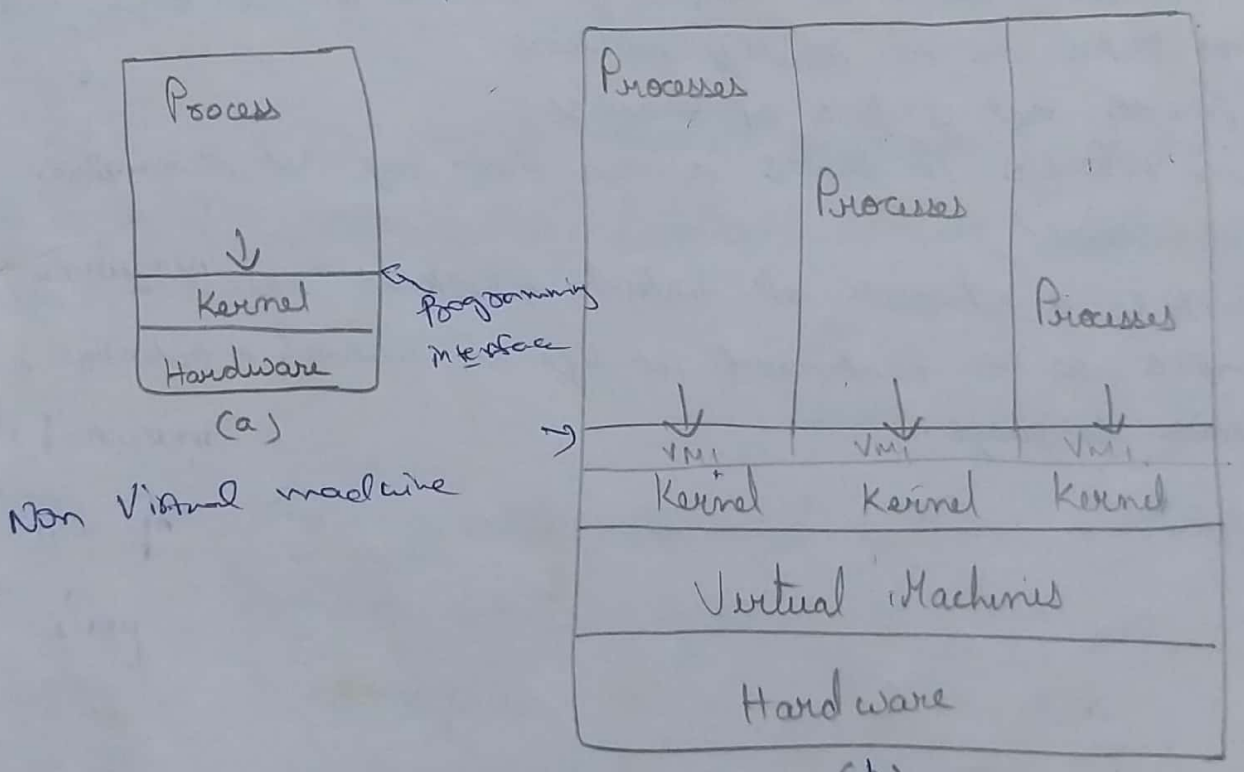
Virtual Machines :

→ Virtual memory techniques an operating system can create of multiple processes, each executing on its own processor with its own memory (virtual)

→ Virtual Machine does not provide any additional function rather provides an interface that is identical to hardware.

→ Each process is provided with a (virtual) copy of the underlying computer

System models



(a) Non Virtual machine

(b) Virtual machine

→ The resources of the physical computer are share to create the virtual machines.

→ A major difficulty with the virtual machine approach involves the disk systems. Suppose that the physical machine has three disk devices but want to support seven virtual machines

→ Users are given their own virtual machine. They can run any of the operating system or s/w packages that are available on the machine.

→ The virtual machine software is concerned with multi programming multiple virtual machine onto a physical machine but does not need to consider any user-support software.

Implementation :-

→ The virtual machine concept is useful it is difficult to implement

→ The machine has two modes 1) user mode & monitor mode

→ virtual machine s/w can run in monitor mode.

→ The virtual machine itself can execute in only user mode.

→ Actions that cause a transfer from user mode to monitor mode on a real machine & virtual machine.

Benefits :- The virtual machine concepts has several advantages.

1) Each virtual machine is completely isolated from all other virtual machines there are no security problems.

2) To provide sharing two approaches

⇒ It is possible to share a mini disk, by this technique files can be shared

⇒ To define a network of virtual systems research & development

⇒ example to be continued utility of virtual machine involves the Java language.