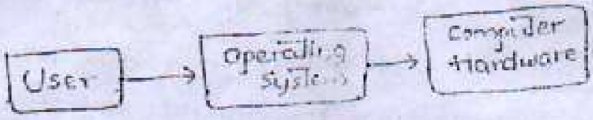
P. Nageswara Rao. M.Tech
CH. Madhavi.

Computer is an electronic device which performs various operations. It does so, by taking data as a input and process the data according to the instruction, provides output in various forms.

INTRODUCTION :-

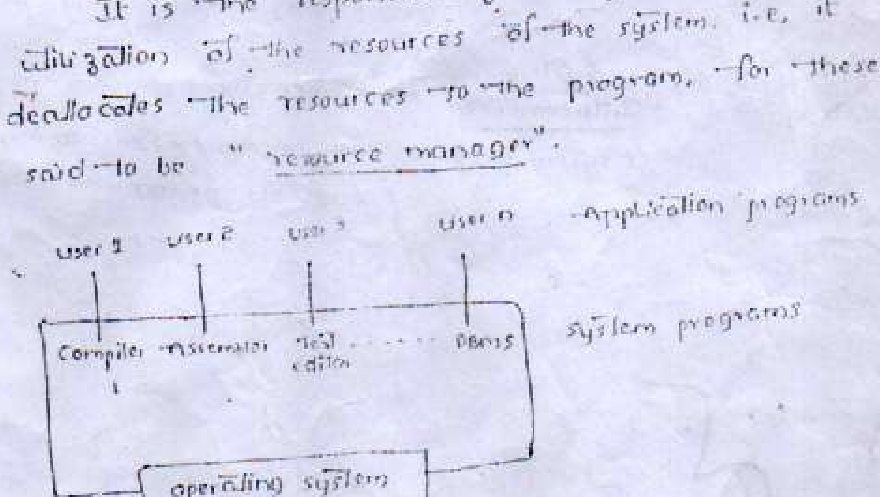
An operating system is an interface between the user and the hardware.

An operating system is an important software which is designed to bridge the gap between the user and the system. An OS utilizes the hardware resources of one or more processors to provide a set of services to the users. The OS also deals with the secondary memory and the input / output devices on behalf of its users.



An operating system controls the computer, manages the system resources and supervises the interaction between the computer system and the user.

It is the responsibility of operating system for allocation and utilization of the resources of the system. i.e, it allocates and deallocates the resources to the program, for these reasons it is said to be "resource manager".



OVERVIEW OF COMPUTER SYSTEM HARDWARE :-

Basically there are four structural elements in computer system. They are :

(i) Processor
(ii) Main memory
(iii) I/o devices
(iv) System interconnection

(i) **Processor** :- The processor is like the brain of the computer. This is the component that controls the operation of the computer and performs its data processing functions. Processor is also referred as microprocessor. When there is only one processor, it is often referred to as the Central Processing Unit (CPU).

(ii) **Main memory** :- The basic requirement of a computer is to store data and programs. This memory is typically volatile i.e. when electric power is cut-off it looses its contents. This memory also referred to as RAM (or) primary memory.

(iii) **I/o devices** :- I/o devices move data between the computer and its external environment. The external environment consists of variety of external devices, including secondary memory devices, communication equipment and terminals. Input devices send data and instructions to the computer and output devices take information out of the system and give it back to the user or to other system.

(iv) **System Interconnection (system bus)** :- Some structure and mechanisms that provides for communication among processors, main memory and I/o devices.

OPERATING SYSTEM OBJECTIVES AND FUNCTIONS :-

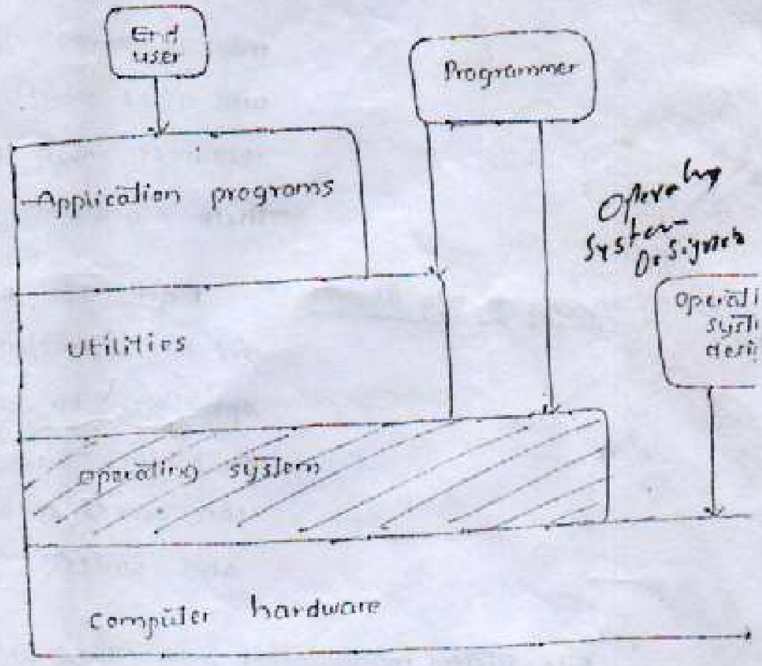
An operating system highlights on three functions or maintains three main objectives :

(i) Convenience :- An operating system makes a computer more convenient to use.

(ii) Efficiency :- An operating system allows the computer system resources to be used in an efficient manner.

(iii) Ability to evolve :- An operating system should be constructed in such a way as to permit the effective development, testing and introduction of new system functions without interferring with service.

Now, we will discuss the three above main aspects of operating system :



The hardware and software that are used in providing applications to the user can be viewed in a layered hierarchical fashion. The one who uses application is called an and is generally not concerned with the architecture of the comp

The programmer concentrates on developing the application program making use of the resources from the utilities which are also used in file management and control of I/o devices

Operating system is the most important system program. It acts as mediator, making it easier for the programmer and for application programs to access and use those facilities and services.

Operating system typically provides services in the following areas :

Program development :- The operating system provides a variety of facilities and services, such as editors and debuggers, to assist the programmer in creating programs.

Program execution :- A no. of tasks need to be performed to execute a program. Instructions and data must be loaded into main memory, I/o devices and files must be initialized and other resources must be prepared. The os handles these scheduling duties for the user.

Access to I/o devices :- Each I/o device requires its own peculiar set of instructions or control signals for operation. The os provides a uniform interface that hides these details so that programmer can access such devices using simple read and writes.

File system manipulation :- The programs may need to read and write files the file system of os deals with reading and writing files in the programs. It also deals with creating and deleting files by name.

Error detection :- The OS should always be aware of errors which may occur in the CPU and memory hardware (such as power failure or memory error). Errors may also occur in I/O devices (such as lack of paper in printer, a connection failure in the n/w) or in the user program (such as attempting to access an illegal memory location or an arithmetic overflow). For each type of error, the OS should take appropriate action to ensure correct and consistent computing.

Resource allocation :- when there are multiple users or multiple jobs running at the same time, resources must be allocated to each of them so that all the jobs are executed efficiently.

Accounting :- Suppose, we want to find out which user utilizes how much and what kind of computer resources. This record keeping can be used for accounting so that the users can be billed based on resources utilized or simply for computing usage statistics. Usage statistics may be a valuable tool for the researchers who wish to reconfigure the system to improve computing services

Protection :- The owners of the information stored in multi-user environment may want to control its use, so that unauthorized persons or users do not have access to the information that they own because the info. may be confidential. when several disjoint processes execute concurrently, it should not be possible for one process to interfere with other. Protection involves ensuring that all access to system resources is controlled.

Security can be implemented by giving each authorized user of a system a password which he/she uses to authenticate himself/herself to the system, to be allowed

EVOLUTION OF OPERATING SYSTEM :-

With the earliest computers. from the late 1940s to the mid-1950s, the programmers interacted directly with the computer hardware; there was no operating system. These machines were run from a console consisting of display lights, toggle switches. some form of input device and a line printer. If an error halted the program, the error condition was indicated by the lights. The programmer could proceed to examine registers and main memory to determine the cause of error. If the program proceeded to a normal completion, the output is appeared on the printer.

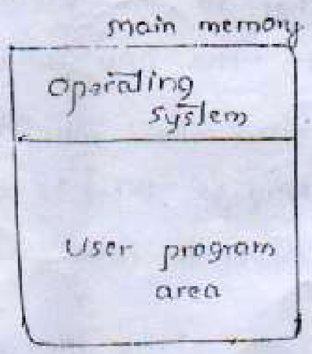
These early systems came up with two main problems:

* Scheduling :- Almost all installations employed a sign-up sheet to reserve machine time. Typically, a user could sign-up for a slot of time in multiples of half hour or so. One might sign-up for an hour and finish in 45 mins. This would result in wasted computer idle time. On the other hand, the user might run into problems, not finish in the alloted time and the be forced to stop before resolving the problem.

* Set-up time :- A job could consist of loading the compiler plus the high level language program into memory, saving the compiled program and then loading and linking together the object program and common functions. Each of these steps could involve mounting or dismounting tapes. If an error occurs, the unlucky user has to go back to the beginning of the set-up sequence. Thus, considerable amount of time will be spent just in setting up the program to run.

Simple Batch systems :-

In earlier times, computers were physically enormously large machines run from a console. The common input devices were card readers and tape drives. The common output devices were line printers, tape drives and card punches. The users of such systems did not interact directly with the computer system. Rather, the user prepared a job - which consists of the program, the data and some control information about the nature of the job (control cards) - and submitted it to the computer operator. The job would usually be in the form of punched cards. At some later time (perhaps minutes, hours or days), the output appeared. The output consisted of the result of the program, as well as a dump of memory and registers in case of program error.

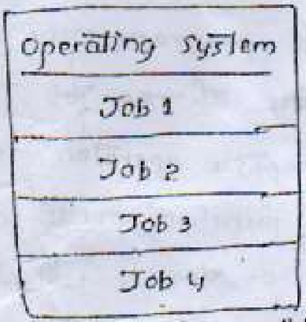


The os in these early systems was fairly simple. Its major task was to transfer control automatically from one job to the next. To speed up processing, jobs with similar needs were matched together and were run through the computer as a group. The operator would sort programs into batches with similar requirements and as the computer became available would run each batch. The output from each job would be sent back to the appropriate programmer.

The delay between job submission and job completion is called Turn around Time

Multiprogrammed Batch Systems :-

In simple batch systems, at a time we can be able to work with a single program. If such program consists many I/O operations, the cpu has to sit idle. In order to avoid the idleness of the cpu, multi programming concept has been introduced.



memory layout for multiprogramming system

In a multiprogramming system, the operating system simply switches to and execute another job. when that job need to wait, the cpu switched to another job and so on. Eventually, the first job finishes waiting and gets the cpu back. As long as there is always some job to execute, the cpu will never be idle.

This idea is common in other life situations. A lawyer does not have only one client at a time. Rather, several clients may be in the process of being served at the same time. While one case is waiting to go to trial or to have papers typed, the lawyer can work on another case. If she has enough clients, a lawyer never needs to be idle. (Idle lawyers tend to become politicians)

Multi programming is the first instance where the OS must make decisions for the users. multi programmed OS are therefore fairly sophisticated.

Time sharing systems :-

Multiprogrammed batched systems provide an environment where the various system resources (for example, cpu, memory, peripheral devices) are utilized effectively

Time sharing or multitasking, is a logical extension of multi programming. multiple jobs are executed by the cpu switching between them, but the switches occurs so frequently that the users may interact with each program while it is running.

In multiprogramming system, the cpu switches from one job to another when an I/o or event occured, where as in time sharing system, the cpu switches from one job to another when "time slice" or "time quantum" completed.

Distributed systems :- (loosely coupled systems)

A distributed system is a collection of processors that do not share memory or a clock. Instead, each processor has its own local memory and the processors communicate with each other through various communication lines, such as high-speed buses or telephone lines.

There are various characteristics that are responsible for usefulness of distributed systems. Some of these are :

→ resource sharing
→ computation - speed up
→ reliability
→ communication

Resource sharing :- When a user at one site tries to access or utilize the resources available at the other, where a no. of different sites are connected to each other, then in this scenario, the resource sharing provides techniques for sharing files at remote sites, processing information in a distributed database, printing files at remote site.

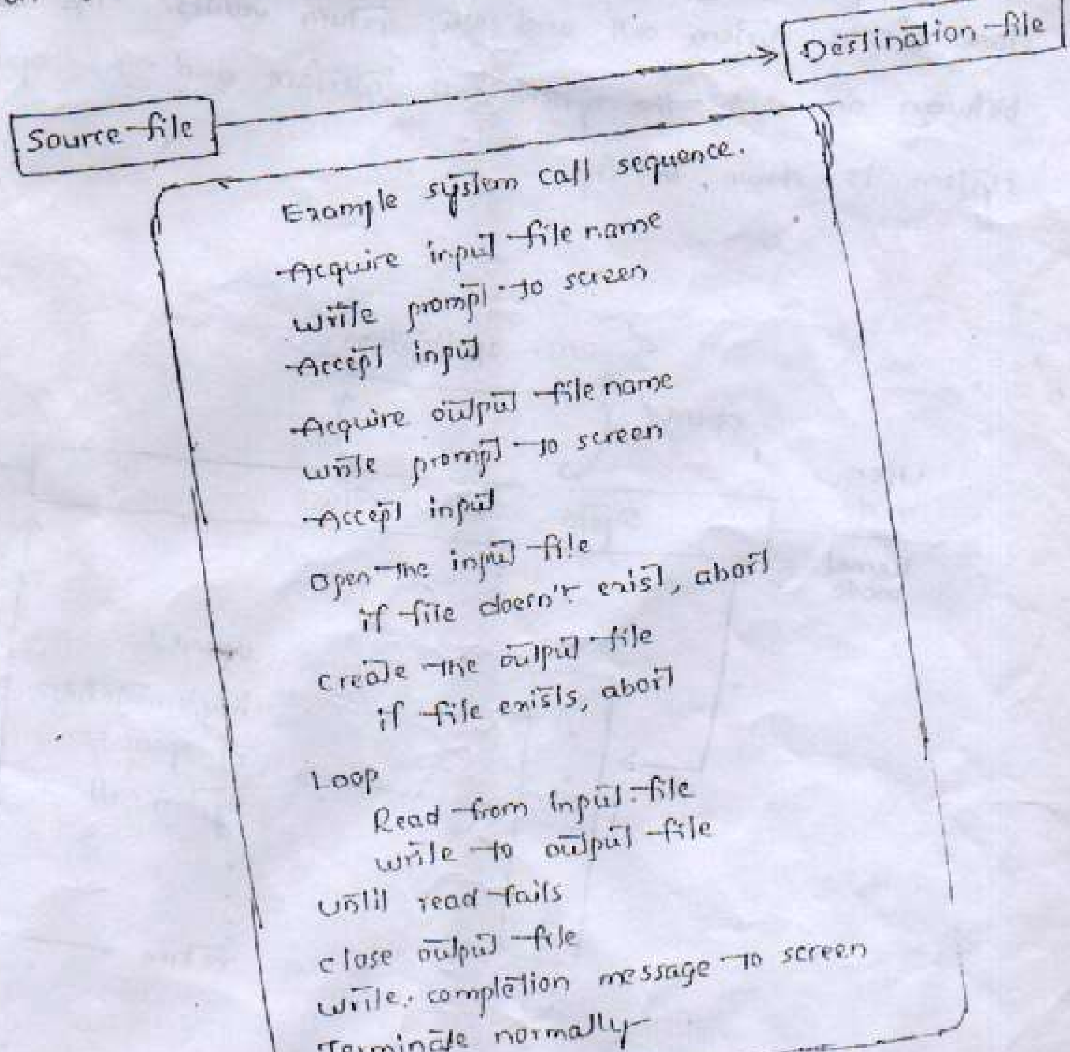
Computation speed-up :- If a particular computation can be partitioned or divided further into a no. of sub computations that can concurrently be loaded, then the distributed system may allow us to distribute the computation among the various sites to run that computation concurrently.

Reliability :- Due to any reason, if it happen that a site fails, then the remaining sites can potentially continue operating.

Communication :- In many situations, programs need to exchange data with each other on one system, users may initiate file transfer or communicate with one another.

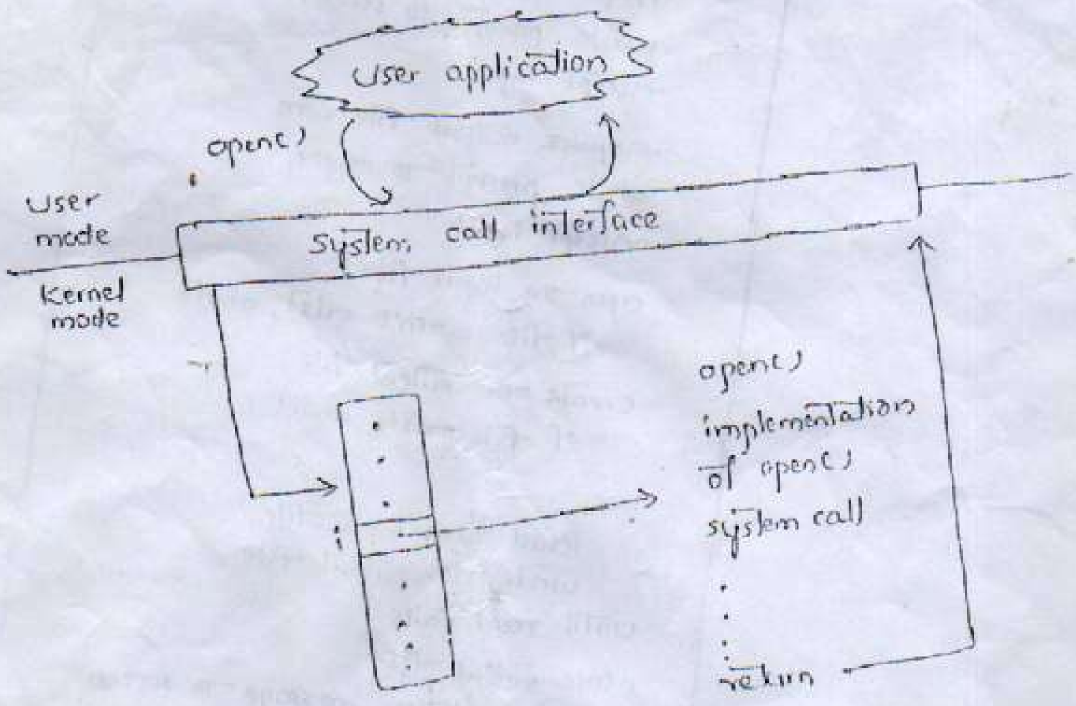
System Calls :- The operating System provides a wide range of system services and functionalities. These services can be accessed by using system calls. The system calls acts as The interface b/w user applications and operating system services. They are available as built-in functions or routines in almost all high-level languages such as C, C++ etc.

Consider an example of copying contents of one file to another. To handle this task, several system calls are used from the time of prompting users to enter file names of source and destination files to copying the contents and closing these files.



Example of how system calls are used

The runtime support system (a set of functions built into libraries included with a compiler) for most programming languages provides a system call interface that serves as the link to system calls made available by the operating system. The system call interface intercepts functions calls in the API and invokes the necessary system call with in the operating system. Typically, a number is associated with each system call and the system call interface maintains a table indexed according to these numbers. The system call interface then invokes the intended system call in the operating system and returns the status of the system call and any return values. The relationship between an API, the system call interface and the operating system is shown in fig:



The handling of user application invoking open() system call

System calls are categorized into 5-types: They are: (13) (4)

- Process control
- File management
- Device management
- Information maintenance
- Communication

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

File management :-
- create file, delete file
- open, close
- read, write, reposition
- get file attributes, set file attributes

Device management :-
- request device, release device
- read, write, reposition
- get device attributes, set device attributes
- logically attach or detach devices

Information maintenance :-

- get time or date, set time or date
- get system data, set system data
- get process, file or device attributes
- set process, file or device attributes

Communications :-

- Create, delete communication connection
- Send, receive messages
- transfer status information
- attach or detach remote devices

⟹ Operating system Generation :-

Operating system generation refers to the process of specifying configuration of specific machine or site where operating system has to be deployed and run. This process is often called as "SYSGEN". The operating system is usually stored floppies or DVD-Roms, CD-Roms and distributed. The "SYSGEN" program is used to generate the operating system. It prompts the system operator to enter information about its hardware configuration or it may read the same from a file and automatically verifies that configuration is available or not.

The following information has to be verified :

- The type of CPU or processor used, the various components it has including instruction set, floating point arithmetic it supports. If there are multiple processors then each of them has to be defined.

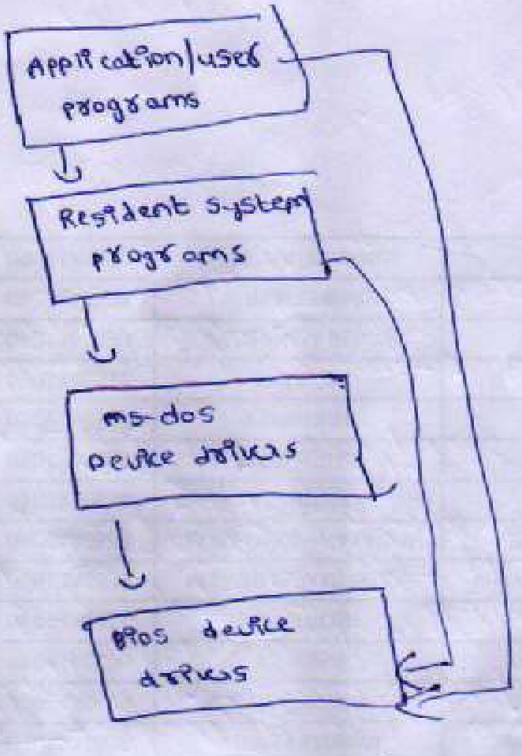
- The memory capacity is allows.
- The various devices it supports and how it addresses them. Some systems maintain device number, Interrupt number, type, model and special characteristic information for each device.
- Finally, the various options that user wants from operating system like size of buffers, cpu-scheduling algorithms to be used, the no. of processes supported at a time, etc. Usually, modern day operating system decide themselves the type of scheduling algorithms to be used and buffer sizes, etc. These are dynamically changed according to system performance.

After getting the above information, O.S. is compiled and data declarations, constant initializations are done to produce a version of O.S. as desired by the user or particular system or machine.

→ Operating System Structures:

- Simple structure
- layered "
- micro kernal "
- Module based "

} Prepare notes on your own



i, structure of ms-dos

i, simple structure:

There are several commercial operating systems which have simple but not well defined structures. usually these systems were developed as small, simple and having limited functionalities with limited space. Its structure was not divided carefully into modules. ms-dos has always experience the threat of vulerable and malicious programs which can cause damage to the entire system. because of improper separation between interfaces and their functionalities. Any application program can access the system or its hardware.

The earlier version of unix also falls in this category. It divides the system in to two parts.

1. kernal
2. System programs

The kernal contains several device drivers which interacts with the system directly. Then the problem occurs in developing kernal. Because it become larger to implement as it has more functionalities.

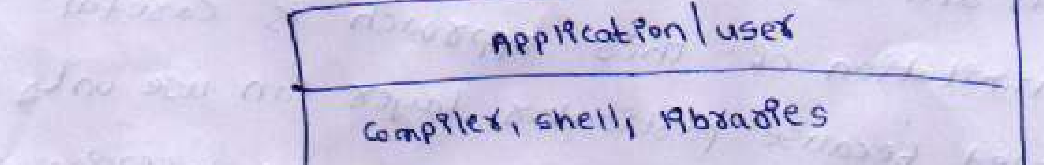
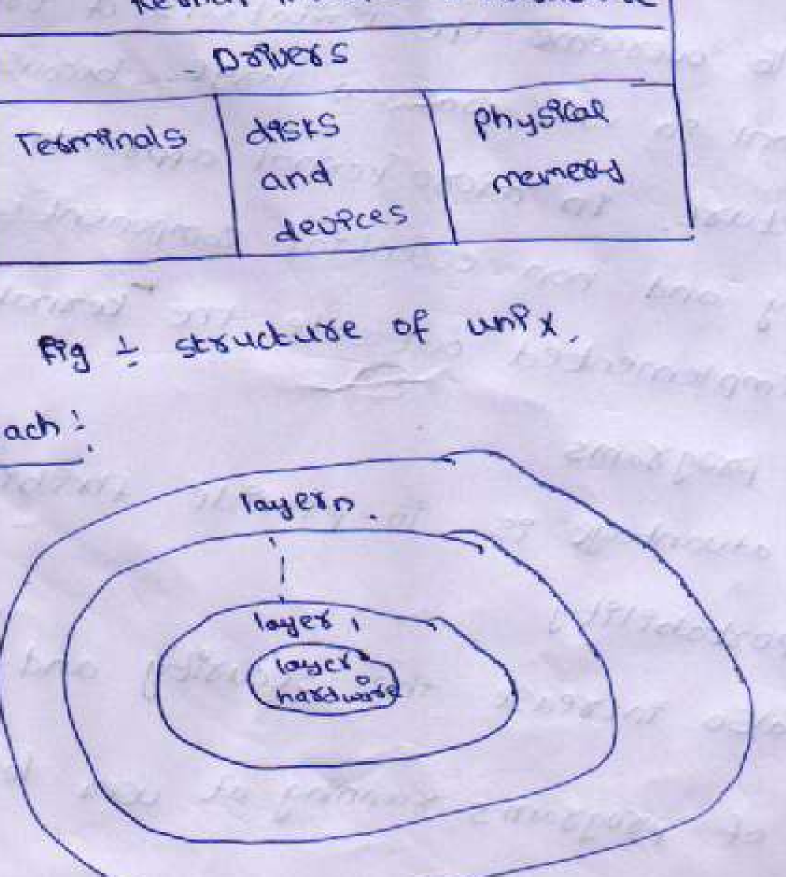


Fig 1: structure of unix.

2. Layered approach:



In this the o.s divided into a number of layers, The highest level corresponds to users or application programs and lowest level corresponds to hardware.

Each layer consist of data structures and operations which are invoked by higher layers, lower layer provided services to the higher level. The Advantage of this approach is that construction and bebugging become simple

The limitation of this approach is careful preplanning is needed because particular layer can use only services of lower layer. Another major problem is inefficiany as it increases the overall burder of o.s

3. Micro kernal:-

it is used to overcome the limitations of traditional unix kernal, the kernal in unix was so large. because of its monolithic structure. In micro kernal approach removes all the unnecessary and non essential components from kernal. These can be implemented out side the kernal as application level or system programs

The advantage is to provide flexibility and extendibility, portability

It also increase the security and protection because most of programs running at user level instead of kernal

In this the o.s divided in to a number of layers, The highest level corresponds to users or application programs and lowest level corresponds to hardware.

Each layer consist of data structures and operations which are invoked by higher layers, lower layer provided services to the higher level. The Advantage of this approach is that construction and bebugging become simple

The limitation of this approach is careful preplanning is needed because particular layer can use only services of lower layer. Another major problem is inefficiency as it increases the overall burder of o.s

3. Micro kernal:-

it is used to overcome the limitations of traditional unix kernal, The kernal in unix was so large. because of its monolithic structure. In micro kernal approach removes all the unnecessary and non essential components from kernal. These can be implemented out side the kernal as application level or system programs

The advantage is to provide flexibility and extendibility, portability

It also increase the security and protection because most of programs running at user level instead of kernal