mood-book

UNIT-I

. An operating system is a program that manages computer hardware. It also provides a basis for application programs and acts as an intermediary between the Computer user and the computer hardware.

→ An operating System is a program that acts as an interface between user and computer hardware & Controls the Execution of all kinds of programs.

→ The two main goals of an operating system (OS) are

* Makes the computer system convenient to use.

* Uses the computer hardware in an efficient manner.

A Computer System can be divided roughly into four components:

1) Hardware 2) the OS 3) the application programs & 4) the Users

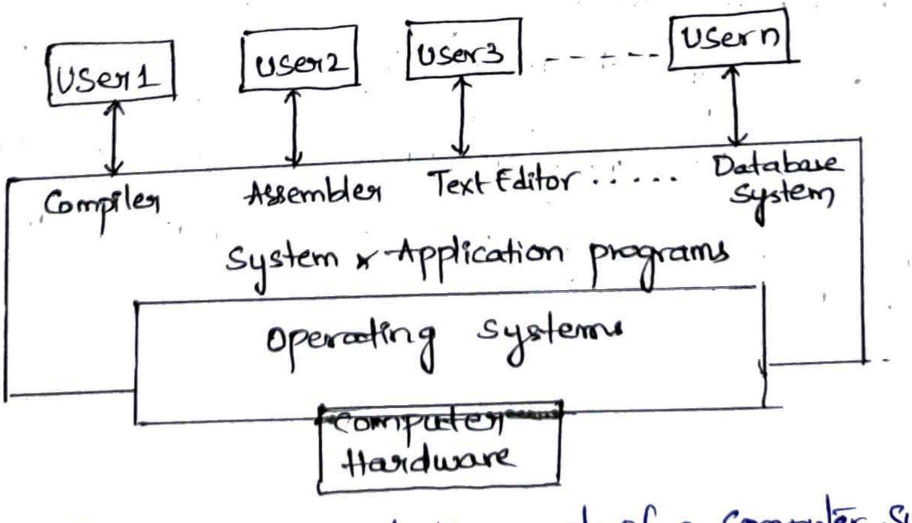


fig. Abstract view of components of a computer system

1) Hardware : the CPU, the memory & the I/O devices provides the basic computing resources for the system.

2) The OS : It acts as a manager for these resources. It controls & coordinates the use of the hardware among various application programs for the various users.

3) The Application programs : Such as word processors, spreadsheets, compilers & web browsers define the ways in which these resources are used to solve users computing problems.

A computer system consists of hardware, software & data. The operating system provides the means for proper use of these resources

⇒ Two views of OS : User's view
System view

* The User view of computer refers to the interface being used.

In personal computer (where single user) operating system is designed for Ease of use with good performance.

In mainframe (where many users access same computer) operating system is designed to maximize resource allocation

In workstations (where users have own resources but also share), the operating system is designed to compromise individual usability & resource utilization.

* from System view, operating system is the program most intimately involved with the hardware. i.e., OS can be viewed as a resource allocator. A computer system has many resources that may be like hardware, software, I/o devices. The operating system acts as the manager of these resources, So that it can operate computer system efficiently & fairly.

The OS need to control the various I/o & user programs. An os is a control program that manages the execution of user programs to prevent errors & improper use of the computer.

⇒ The fundamental goal of computer systems is to Execute user programs & to make solving user problems easier. Toward this goal, computer hardware is constructed. Since bare hardware is not enough, application programs are developed. these programs require certain common operations. These common functions of controlling & allocating resources are brought together into one piece of software: the operating system.

Kernel The OS is the one program running at all the times on computer. The kernel is the central part of an OS. It manages the operations of the computer & the hardware most notably memory & cpu time.

Booting : Start of computer & put it into a state of readiness for operation.

Cold booting : When Computer is started after having being switched off

Warm booting : When the System is restarted

The OS concentrates on optimal use of computing resources. OS has evolved over time.

Eg:- UNIX started with a Keyboard & printer as its interface. Overtime, H/w changed & UNIX was imported to new hardware with more user-friendly interfaces. Many GUI were added, allowing UNIX to be more convenient for users while still concentrating on efficiency.

→The design of an OS is an complex task.

The OS has 3 main objectives:

Convenience : OS makes a computer more convenient to the user.

Efficiency : An OS allows the computer system resources to be used in an efficient manner to ensure good resource utilization & provides appropriate corrective actions when it becomes low.

Ability to evolve : An OS should be constructed in such a way to permit the effective development, testing of new system functions without interfering service.

Memory Hierarchy:

In Computer Systems memory can be used to store data & programs. All the memory devices are put in a hierarchical order based on following criteria:

* Size of memory device
* Accessing speed of Memory device
* Cost per one-bit of Memory

→ Memory devices can be classified based on volatile ability into two categories:

* Volatile devices :
  In this, data can be lost when power supply is off.

* NON-volatile devices:
  In This data won't be lost when power supply is off.

→ Memory devices can be classified into two categories based on accessibility:

* Random Access Devices
  In this, data can be directly accessed from memory device

* Sequential Access Devices
  In this data can be accessed in a sequential manner

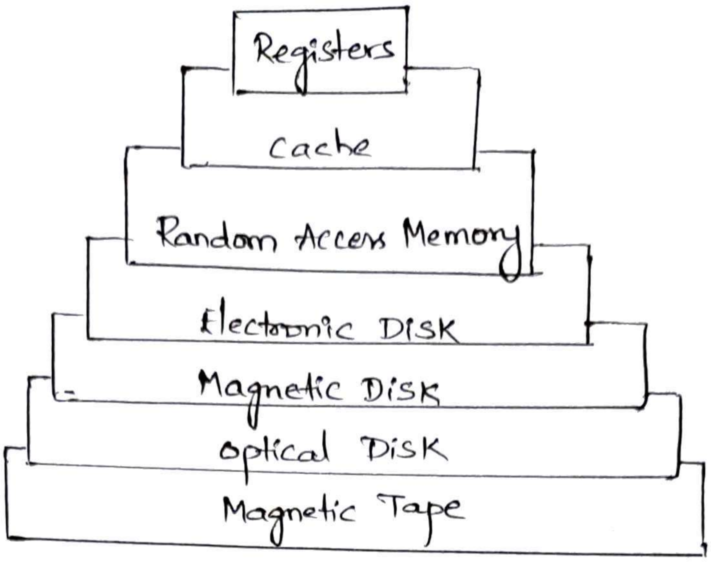


Fig: Memory Hierarchy

Registers :

Registers are used by the cpu for its internal purpose while Executing programs. In registers there are two types

* General purpose Registers

Used to store. input data while performing arithmetic operations.

Eg: Accumulator

* Control Registers

Used to store status of the program Execution.

Eg:- PC, IR, MAR, MDR

Cache Memory

It is placed between registers & main memory. It is used to store data recently used by the processor. It increases accessing speed.

Random Acces Memory:

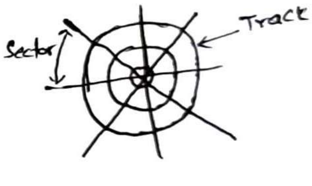
It is also called as Main Memory. It is used to store program and data from secondary memory. In this data can be accessed directly by specifying the address.

Electronic DISK:

It is also called as flash Memory. it acts as RAM until the power Supply is ON. Once the power supply is off it uses internal back up (battery) to store The current data in magnetic disk. So, it is called as Volatile & Non-volatile memory device.

Magnetic Disk:

It is also called as Secondary Memory. It is used to store all the programs. It is divided into sectors & tracks. It is non-volatile device. In general, it is used to provide battery



Optical Disk:

It is called as Removable memory. It is used to carry data along with user. It is manufactured using optical technology.

Eg:- CD-ROM, DVD

Magnetic Tape :

It is the oldest form of memory device. On this, the data can be read/write in a sequential Order. In this, the tape can be divided into two tracks horizontally & stores data in track1 & track2 seperately. Track1 data can be accessed in forward direction & track2 data can be accessed in reverse direction.

Computer Architecture

A computer's system Architecture can be categorized depending upon the availability of general purpose processors.

- * Single processor systems
- * Multi processor systems

Single processor Systems :

On a single processor system, there is one main cpu capable of Executing a general ~~processor~~ purpose instruction set, including instructions from user processors.

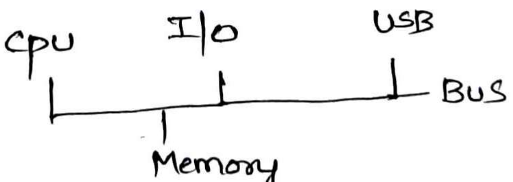


fig. Single processor system

Low Reliability, less throughput

Resource sharing is not possible

More cost.

Multi-processor Systems:

On a multi-processor system there will be two or more number of general purpose processors, sharing the computer bus sometimes the clock, memory & peripheral devices

→ Multiprocessor systems have:

Increased Throught

Increased Reliability

Economy of Scale

→ There are two types of multi processor systems

symmetric Multi-processor systems (SMP)

Asymmetric Multi-processor systems

Symmetric Multiprocessor system:

In these systems Each processor performs all tasks within the operating system. i.e., all processors are peers; no master-slave relationship Exists between processors.

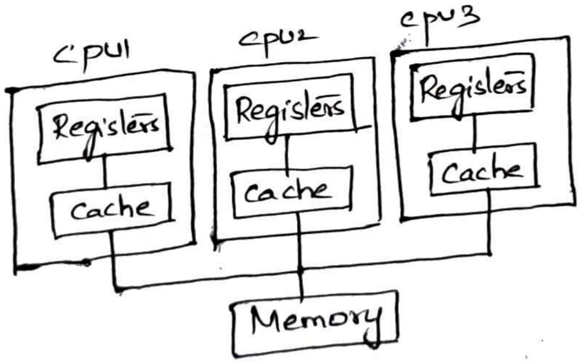


fig: SMP Architecture

Asymmetric Multiprocessor systems:

One among all the processors acts as Master other cpu acts as slaves & performs the operations instructed by master & at the same time monitors the master processor. The Master processor schedules & allocates work to the slave processors.

Multicore systems:

Multiple computing cores on a single chip. They can be more efficient than multiple chips with single core because on-chip communication is faster than between-chip communication. Each core will have its own register set also cache.

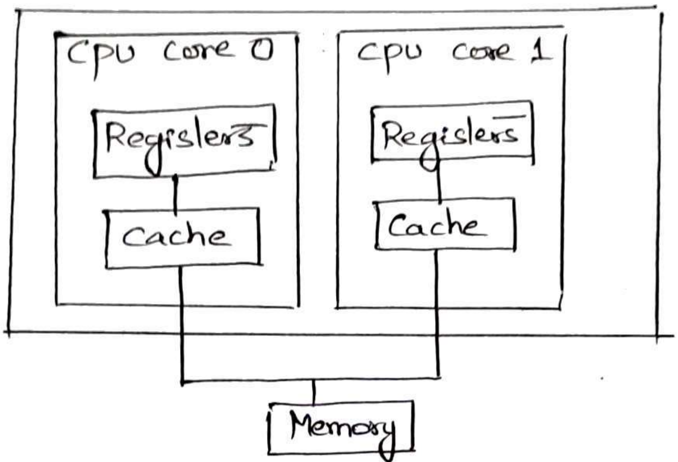


Fig. Dual Core Design with two Cores on single-chip

clustered Systems:

These gather together multiple cpu's to accomplish computational work. They are composed of two or more individual systems. These systems share storage & are closely linked via local-area-network.

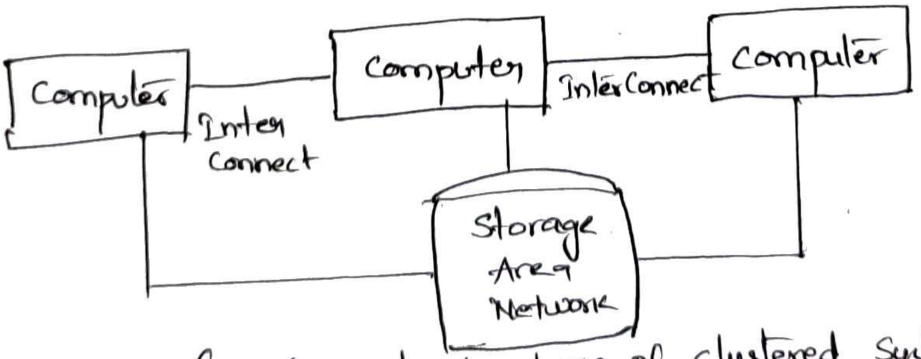


fig. General structure of clustered systems

Mainframe Systems:

Mainframe compiter systems were the first computers used to tackle many commercial & scientific applications.

Here we discuss the growth of mainframe systems from simple batch systems, where the computer runs one & only one application to time-shared systems, which allow for user interaction with the computer system

① Simple Batch Systems:

→ Early computers were physically enormous machines run from a console.

→ The common input devices were card readers & tape drivers.

→ The common output devices were line printers, tape drives & card punches.

→ The user didnot interact directly with the computer systems.

→ The user prepared a job, which consists of the program, the data & some control information about the job & submitted it to the computer operator in the form of punched cards.

→ At some later time (after minutes, hours or days) the output appeared.

The output consisted of the result of program, as well as a dump of final memory & register contents for debugging.

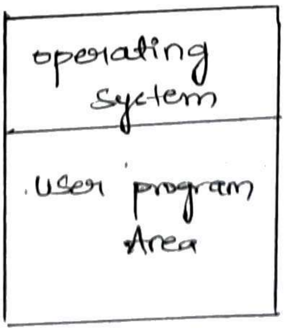


fig. Memory layout for a simple batch system

→ The OS in the early computer were simple its major task was to transfer control automatically from one job to the next.

→ The OS always resides in memory.

→ To speedup processing, operators batched together jobs with similar needs & ran them though 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.

→ In this Execution environment, the cpu is often idle because the speeds of the mechanical I/o devices are very slow than those of electronic devices.

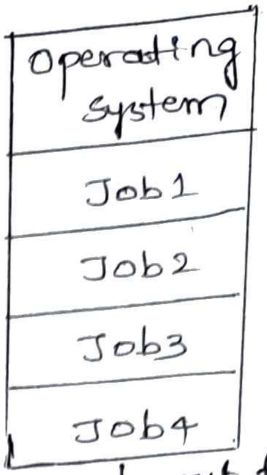
→ Even a slow cpu executes thousands of instructions per second. A fastcard reader on the otherhand might read 20 cards per second. Thus, the difference in speed between the cpu & its I/o devices may be 3 order of magnitude or more.

→ Overtime, improvements in technology & the introduction of harddisks resulted in faster I/o devices. The introduction of disk technology allowed OS to keep all jobs on a disk with direct access to several jobs, the OS could perform job scheduling.

(2) Multiprogrammed systems:

A single user cannot keep either cpu or I/o devices busy at all the times.

Multiprogramming increases cpu utilization by organizing jobs so that the cpu always has one to Execute.



Memory layout for a Multiprogramming system

→ The OS keeps serval jobs in memory simultane-ously. This set of jobs is a subset of the jobs kept in the job pool.

→ The OS picks & begins to Execute one of the jobs in the memory. Eventually, the job may have to wait for some task such as an I/o interrupt operation to computer.

→ In a non-multiprogrammed system, the cpu would sit idle.

→ In a multiprogrammed system, the OS simply switches to and Executes another job when that job needs to wait, the cpu is switched to another job & so on.

→ Multiprogrammed OS are sophisticated. All the jobs that enter the system are kept in the job pool. This pool consists of all process residing on disk awaiting allocation of main memory. If several jobs are ready to be brought into memory & there is no sufficient space, then the system chooses by making

(3) Time-sharing systems:
(or) Multi-tasking systems

Multiprogrammed systems provides an environment where various system resources (cpu time, Memory, I/o devices) were utilized effectively, but it didnot provide user interactions with computer.

→ Time sharing or Multitasking is an logical Extension of multi programming.

→ The cpu Executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with Each program while it is running.

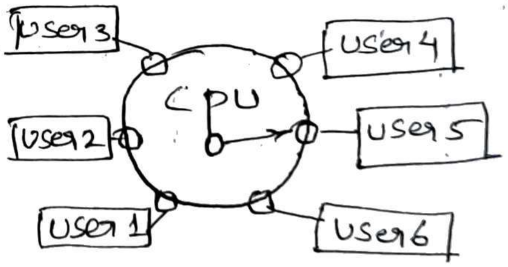
→ An interactive computer system provides direct communication between the user and the system.

→ The user gives instructions to the operating system or to a program directly, using a keyboard or mouse & waits for immediate results.

→ Therefore, the response time of the system should be short.

→ A time sharing operating system allows many users to share the computer simultaneously.

→ Since Each action or command in a time-shared system leads to be short, only a little cpu time is needed for Each user,



As the system switches rapidly from one user to next, Each user is given the impression that the entire computer system is dedicated to use, Even though it is being shared among many users.

A time shared operating system uses CPU scheduling and multiprogramming to provide Each user with a Small portion of a time-shared Computer.

process : A program under Execution

(or)

- A program loaded into memory & Executing is Commonly referred as a process.

→ Time sharing OS are Even more complex then multiprogrammed OS.

→ In both, Several jobs must be kept simultaneously in memory, so the system must have memory management & protection.

→ To Obtain good response time, jobs may have to be swapped in & out of main memory to the disk. The disk acts as a backing store for main memory and it is achieved by the method Virtual memory that allows the Execution of a job that may not be completely in memory.

→ Time sharing systems must also provide a file system that resides on a disk, hence disk management must be provided.

→ Also time sharing systems provide a mechanism for Concurrent Execution, that must provide mechanisms for job synchronization & Communication.

(4) Personal computers | Desktop systems :

→ pc's appeared in 1970's.

→ pc O.S were neither multi user nor multitasking

→ The goal of this OS is to maximize cpu and peripheral utilization i.e, to maximize user convenience & responsiveness.

Eg:- pc's running Microsoft Windows, Apple Macintosh, MS DOS - upgraded to OS/~~to~~ multitask system

Linux (a UNIX like OS) for pc's also became popular.

→ Microcomputers were able to adopt some of the technology developments. On the other hand, hardware costs for microcomputers are sufficiently low that individuals have sole use of the computer & cpu utilization: is no longer a prime concern.

→ File protection at first was not necessary on a personal computer. But now-a-days these computers are connected to other computers over local area networks (LAN), MAN, WAN, etc.,

→ While other computers & other users can access the files on a pc, file protection again becomes a necessary features of the OS.

→ The Lack of such protection has made it easy for malicious programs (or) virus programs to destroy data on systems such as MS-DOS & macintosh operating system.

Multiprocessor systems / parallel systems / Tightly coupled systems

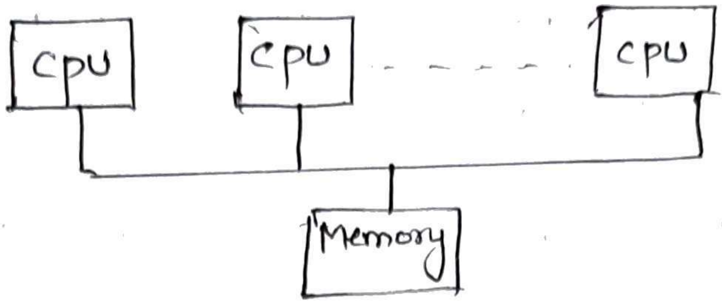


fig. Multiprocessing Architecture

→ Most systems are single-processor systems i.e., they have only one cpu.

→ However, multiprocessor systems are growing day-by-day. Such systems have more than one processor, in close communication sharing the Computer bus, the clock, Sometimes memory & peripheral devices

Multiprocessor systems have 3 main advantages:

1) Increased throughput : Throughput is defined as the amount of work that can be performed (or) the amount of output that can be produced by a system in unit time.

By increasing number of processors, we hope to get more work done in less time. When multiple processors cooperate on a task, a certain amount of overhead in keeping all the parts working correctly plus contention for shared resources lowers the Expected gain from multiple processors.

2) Economy of scale : Multiprocessor systems can save more ~~memory~~. money that multiple single processor systems, because they can share peripherals mass storage & power supplies.

If Several programs operate on the same set of data it is cheaper to store those data on one disk & to have all processors share them, then to have many computers with local diske & many copies of the data.

3) Increased Reliability : If functions can be distributed properly among several processors, then failure of one processor will not halt the system only slow it down.

→ The processors are connected by a bus.

→ The Most common multiple processor systems now use Symmetric multiprocessing (SMP) means that all processors are peers, no master-slave relationship Exists between processors. Each processor concurrently runs a copy of the OS.

→ The benefit of this model is that many processes can run Simultaneously N processes can run if there are N cpu's without causing a significant detoriorartion of performance.

→ A Multiprocessor system will allow processes & resources such as memory to be shared dynamically among various processors.

Distributed Systems:-

- A network is a communication path between two of more systems.
- Distributed systems depend on networking for their functionality.
- Networks are based on the distances between the nodes.

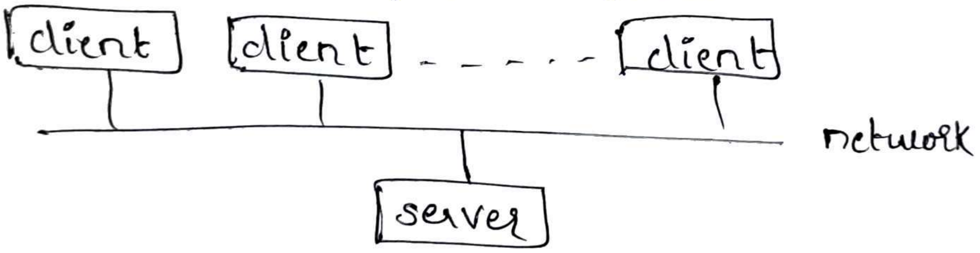
LAN - Local Area Network - exists within a room, a floor, or a building.

MAN - Metropolitan Area Network - links buildings within a city.

WAN - Wide Area Network - exists between building, cities of countries.

① Client server systems:

- Centralized systems today act as server systems to satisfy requests generated by client systems.



Server systems can be categorised as

1) Computer -Server systems:-

Provides an interface to which clients can send requests to perform an action, in response to which they execute the action and send back results to the client.

2) File Server Systems:-

Provide a file system interface where clients can cre update, read and delete files.

(6) Peer - to Peer Systems:

- The computer networks used in these applications consists of a collection of processors that donot show memory.
- Instead each processor has its own local memory.
- The processors communicate with each other through communication lives such as telephone lines, high-speed buses etc.
- These systems are usually refered as loosely coupled systems or distributed systems.

(7) Real- Time systems

- A Realtime system is used when rigid time requirements have been placed on the operation of a processor.
- Sensors bring data to the computer. The computer must analyze the data and possibly adjust controls to modify the sensor inputs.
- Ex:- medical Imaging systems, Industrial control systems, Systems that control scientific experiments, home-appliance controllers weapon systems etc.
- A real time system has a well-defined, fixed time Constraints

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

- Ex: It would not do for a robot arm to be instructed to halt after it had smashed into the car it was building.

- A real time system functions correctly only if it returns all correct result within its time constraints.

SYSTEM COMPONENTS:

- An OS provides the environment within which programs are executed.
- Many modern systems share the goal of supporting the system components.

1) Process Management:

- A process can be thought of as a program in execution.
- A process needs certain resources - like CPU time, memory, files and I/O devices - to accomplish its task.
- These resources are either given to the process when it is created or allocated to it while it is running.
- A system consists of a collection of processes, some of which are Operating system processes (those that execute system code) and the rest of which are user processes (those that executes user code).
- The operating system is responsible for the following activities in connection with process management.

* Creating and deleting both user and system processe.
* Suspending and resuming processes.
* Providing mechanisms for process synchronization.
* Providing mechanisms for process communication.
* Providing mechanisms for deadlock handling.

- A program is a passive entity, such as the contents of a file stored on disk.
- A process is an active entity, with the program counter specifying the next instruction to be executed.

2) Main-Memory Management:

Main memory is a darge array of words or bytes, ranging in size from hundreds of thousands to billion.

- Each word on byte has it own address.
- The CPU reads instructions from main memory.
- For a program to be executed, it must be first loaded into memory. As the program terminates, its memory space is declared available, and the next program can be loaded and executed.
- To improve the CPU utilization and computers respond to its users, several programs are loaded in memory.
- The OS is responsible for the following activities in connection with memory management.

* Keep track of which parts of memory are currently being used and by whom.

* Deciding which processes are to be loaded into memory when memory space becomes available.

* Allocating and Deallocating memory space as needed.

3) File - Management.

- A file is a collection of related information.
- A file consists of a sequence of bits, bytes, lines or records.
- commonly files represent programs and data.
- Data files may be numeric, alphabetic, alphanumeric etc.
- The OS implement the concept of file by managing the storage media such as disks, tapes and the devices that control them.
- Files are normally organized into directories to ease their use.
- Various file operations must be supported such as read, write, append etc.
- The O.S is responsible for the following activities in connection with file management.

1) creating and deleting files.

2) Creating and deleting directories.

3) supporting primitives for manipulating files & directories.

4) Mapping files onto secondary storage.

5) Backing up files on stable storage media.

4) I/o - system Management:

- One of the purpose of an OS is to hide the pecularities of hardware devices from the user by the I/o subsystem.
- The I/o subsystem consists of
  (i) A memory - management component that includes buffering caching and spooling.
  (ii) A general device-driver interface.
  (iii) Drivers for specific hardware devices.

5) Secondary storage Management.

- The main purpose of a computer system is to execute program.
- These programs with the data they access must be in main memory during execution.
- The computer system must provide Secondary storage to back up main memory. Hard disk is used as the main storage medium for both programs & data

The disk is used as the both source & destination for processing.

- Hence, O.S is responsible for proper management of disk storage as follows:

(i) Free-space management.

(ii) Storage allocation.

(iii) Disk scheduling.

6) Network Management :

- A distributed system is a collection of processors that donot share memory, peripheral devices etc.
- The purpose processors in the system are connected through a communication network.
- A distributed system collects physically seperate systems into a single coherent systems, providing the user with access to the various resources that the system maintain.
- Access to a shared resource allows computation speedup, increased functionality, increased data availability and enhanced reliability.
- The innovation of WWW, to create a new access method for information sharing.

FTP → (File Transfer Protocol)

NFS → (Network File system)

7) Protection System:

- If a computer system has multiple users and allows concurrent execution of multiple processes, then the variou processes must be protected from one another activities.
- Protection is any mechanism for controlling the access of programs, processes or users to the resources defined by a computer system.
- Protection mechanisms ensure that the files, memory, CPU and other resources can be operated on by only those processes that have gained authorization from the operating system.
- Protection can improve reliability by detecting errors at the interfaces between subsystems.

8) Command-Interpreter System

- One of the most important system programs for an OS is the command interpreter, which is the interface between the user and the operating system.
- Many commands are given to the OS by control statements.
- A program that reads and interprets control statement is executed automically and is called as the command-line interpreter, and is often known as the shell. Its function is to get the next command statement and execute it.

Ex:- In windows OS mouse is moved to position the mouse pointer on icons. Depending on the mouse pointer location. clicking a button on the mouse, can invoke a program or file or pull down a menu etc.

Ex:- Commands are typed on Keyboard and displayed on a screen with the enter button, signalling the a command is complete and is ready to be execute.
ie, in MS-DOS & UNIX shells operate in this way.

Operating system services:

- An OS provides an environment for the execution of programs.
- It provides certain services to programs and to the users of those programs as shown in figure below.
- These OS services are provided for the convenience of the programmer, to make the programming task easier.

1) Program Execution: The system must be able to load a program into memory and to run that program. The program must be able to end its execution either normally or abnormally.

2) I/O operations: A program may require I/O. This I/O may involve a file or an I/O device. Users usally cannot control I/O devices directly. Therefore OS must provide a means to do I/O.

3) File - System Manipulation: Programs need to read write files. Programs also need to create and delete files by name.

4) communications: In many circumstances, one process needs to exchange information with another process. such communication can occur via shared memory. or by the technique of message passing in which packets

5) Error Detection: The OS constantly needs to be aware of possible errors. Errors may occur in the CPU, memory, I/O devices, user program etc. For each type of error the OS must be take appropriate action to ensure correct & consistent computing.

6) Resource allocation: When multiple users are logged on to the system or multiple jobs are running at the same time, resources must be allocated to each of them.

7) Accounting: we want to keep track of which users use how many and which kinds of computer resources. This records can be used for accounting (or) for knowing the usage statistics. Usage statistics may be a valuable information to improve computing services.

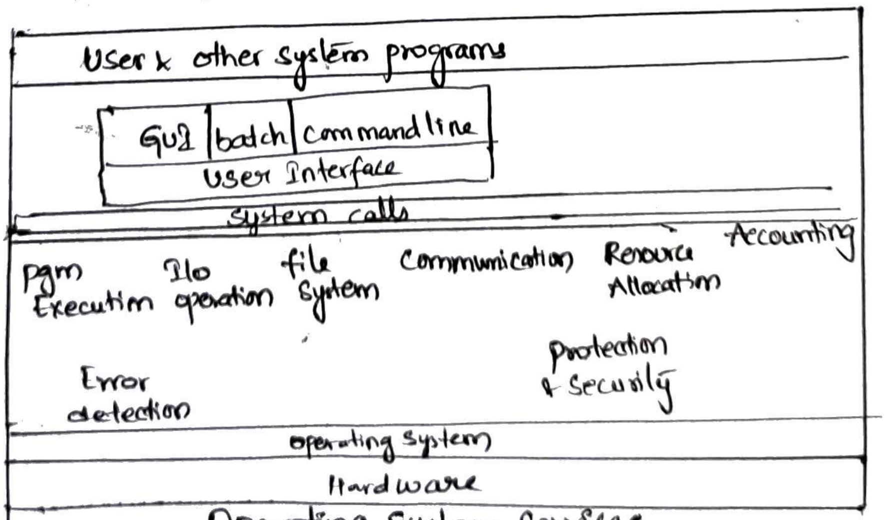
i) Protection: When several processes execute concurrently it should not be possible for one process to inteferе with others, or with the operating system itself. Protection involves ensuring that all access to system resources is controlled.

Security of the system usually means, each user to authenticate himself to the system by a password to be allowed access to the resources. Invalid access attempts are recorded for detection of break-ins.

SYSTEM CALLS:

- System calls provide an interface to the services made available by an OS.
- Ex: To illustrate how system calls are used: Writing a simple program to read data from one file and copy them to another file.

Source file
Destination file



Operating System Services

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 pro completion message to screen
- Terminate normally.

- Every, even simple programs may make heavy use of the O.S. systems execute thousands of system calls per second.

- Each OS. has its own name foe each system call.

application developers design programs according to an application programming interface (API).

- The API specifies a set of functions that are available to an application programmer, including the parameters that are passed to each function and the return values the programmer can expect

Ex:- win32 API for windows Systems.

POSIX API for POSIX- Based systems (includes all versions of UNIX, LINUX & Mac OS X).

Java API for designing programs that run on the Java virtual machine. (JVM).

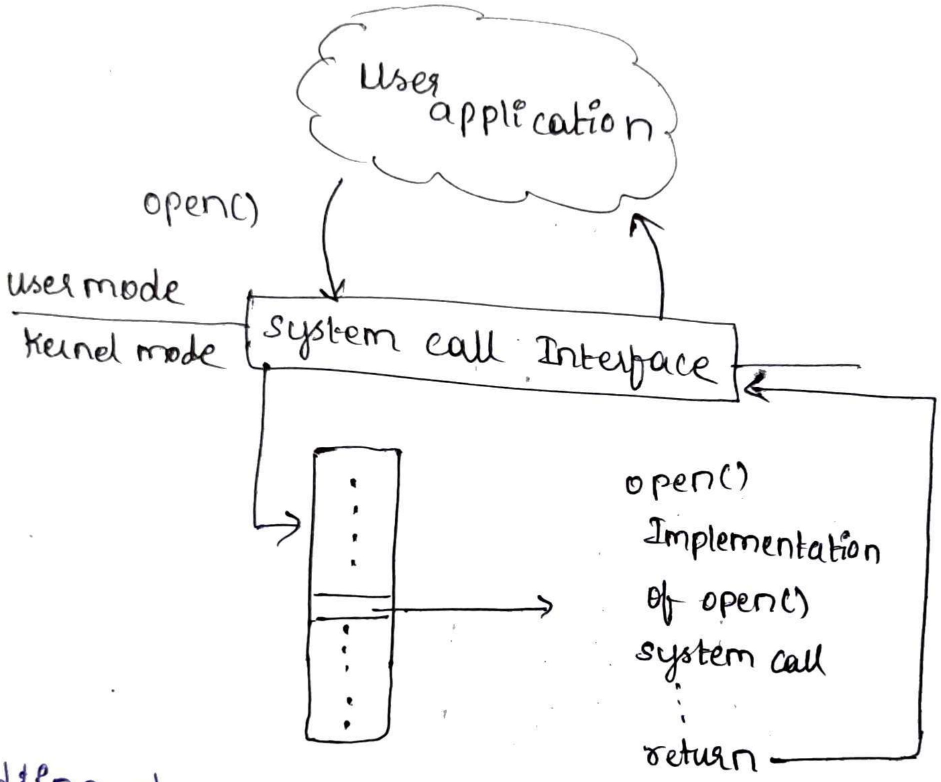
- Each OS has its own name for each system call.

- The benefit of programming according to an API is;

* An application programmer, designing a program using an API can expect his program to compile and run on any system that supports the same API.

* Further, actual system calls can often be more detailed and difficult to work with, than the API available to the application programmer.

- The run-time support system (set of functions bui libraries included with a compiler) for most programming languages provides a system-call Interface, that serves as the links to system calls made available by the OS.
- The system-call interface intercepts function calls in the API and invokes the necessary system call within



Handling of a user application invoking the open() system call.

- The caller needs to know nothing about how the system call is implemented or what it does during execution.
- Thus, most of the details of the OS interface are hidden from the programmer by the API.

ypes of system calls.

ystem calls can be grouped roughly into 5 major categories.

1) Process control

- end, abort
- load, execute
- Create process, terminate process
- get process attributes, set process attributes.
- wait event, signal event.
- allocate and free memory

2) File management

- Create file, delete file.
- open, close.
- read, write, reposition
- get file attributes, set file attributes

3) Device Management

- Request device, Release device
- read, write reposition
- get device attributes, set device attributes
- logically attach or detach devices.

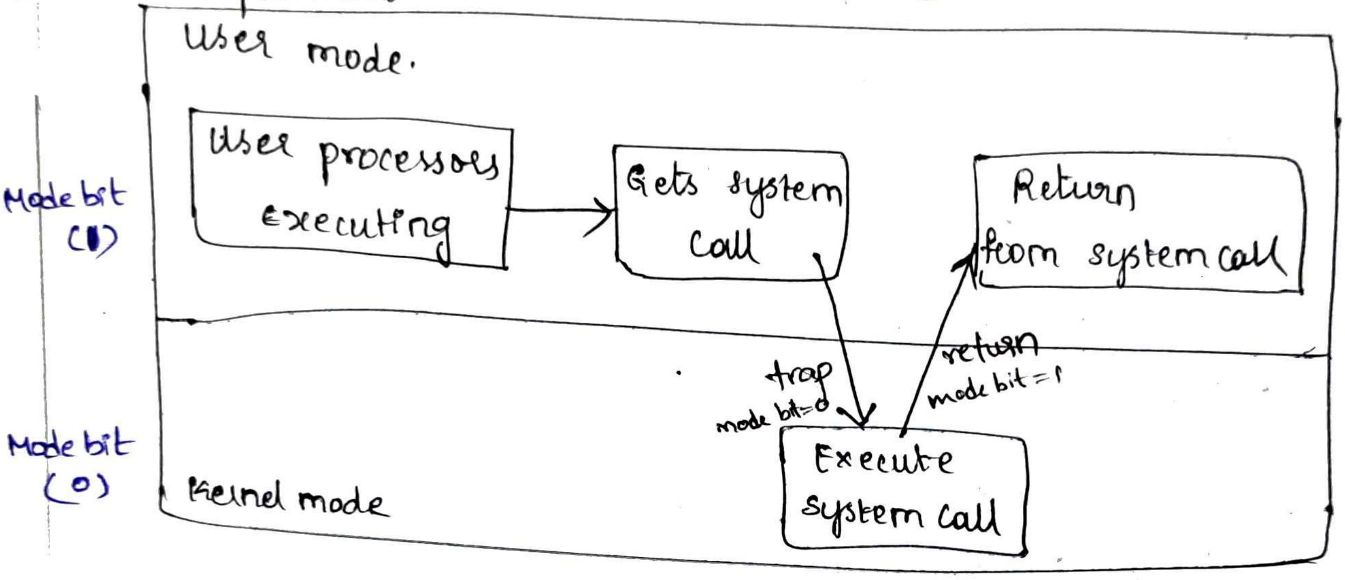
4) Information maintenance

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

5) Communications

- create, delete communication connection.
- send, receive messages.
- transfer status information.
- attach or detach remote devices

Dual Mode operations



Transistion from user to Kernel Mode

Operating System Structure : (Design & Implementation)

A system as large and complex as a modern operating system must be engineered carefully if it is to function properly and be modified easily.

1, Simple structure;

It is written to provide the most functionality in the least space,

It is not divided into modules.

Eg.- MS-DOS

→ The interfaces & levels of functionality are not well Seperated.

→ The Application programs are able to accen the basic I/o routines to write directly to the display & disk drives. This leaves MS-Dos Vulnerable to errant programs, causing entire system crashes when user performs fail

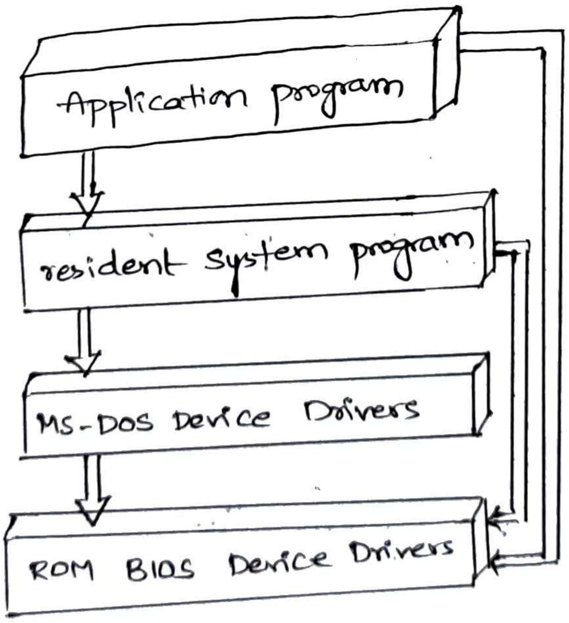


Fig. MS-DOS layer structure

3, Layered Approach :

The Operating System is divided into a number of layers, each built on top of lower layers. The bottom layer is the hardware, the highest (layer N) & the user interface.

→ With modularity, layers are selected such that each use functions (operations) & services of only lower-level layers.

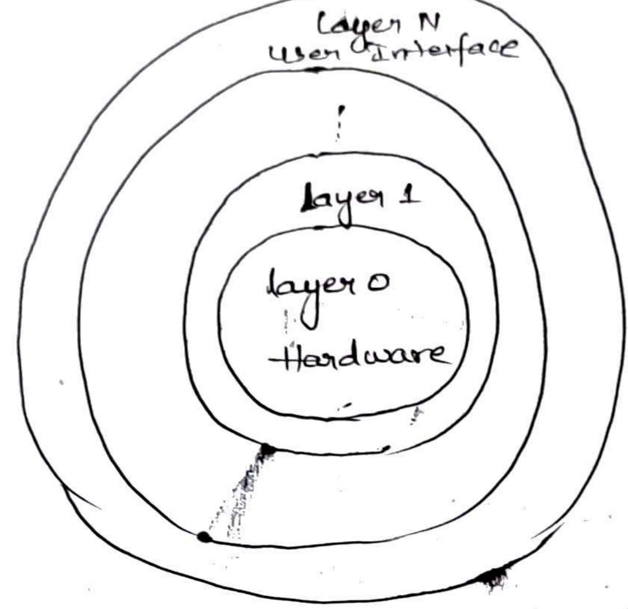


Fig. A layered Operating System

The advantage of the layered approach is simplicity of construction & debugging.

→ The major difficulty with this approach involves appropriately defining the various layers.

→ The approach tends to be less efficient than other types, each layer adds overhead to the system calls.

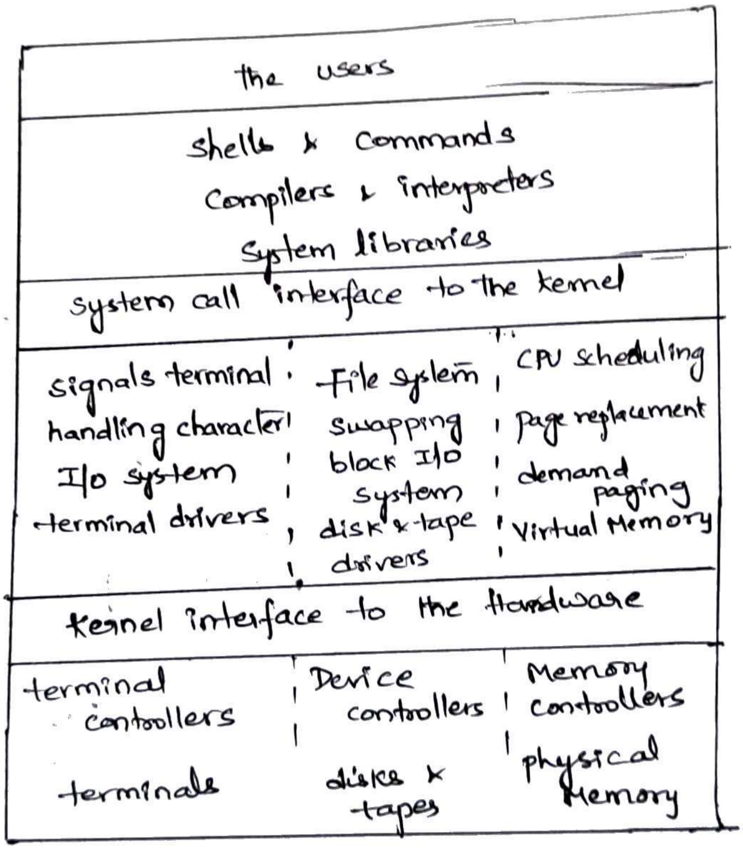
5, Traditional UNIX System Structure :

It consists of two seperable parts :

i, Kernel &

ii, system program

The kernel is further seperated into a series of interfaces & device drivers. The kernel provides the file system, CPU scheduling, memory management & other operating system functions through system calls



the users		
shells & commands compilers & interpreters system libraries		
system call interface to the kernel		
signals terminal handling character I/o system terminal drivers	File system swapping block I/O system disk & tape drivers	CPU scheduling page replacement demand paging Virtual Memory
kernel interface to the Hardware		
terminal controllers terminals	Device controllers disks & tapes	Memory controllers physical Memory

fig: Traditional UNIX system structure

Micro Kernels :

This method structures the OS by removing all non-essential component from the kernel & implementing them as system &

user-level programs. The result is a smaller kernel.

→ There is little consensus regarding which services should remain in the kernel & which should be implemented in user space.

→ Microkernels provide minimal process & memory management in addition to a communication facility

→ The advantage of this approach is ease of Extending the operating system.

4, Modules :

In this approach object oriented programming techniques are used to create a modular kernel.

→ Dynamically loadable modules are used.

→ for Example, the solaris OS structure is organized around a core kernel with seven types of loadable kernel modules.

1, scheduling classes
2, file systems
3, loadable system calls
4, Executable formats
5, STREAMS Modules
6, Miscelleneous modules
7 Device & bus drivers

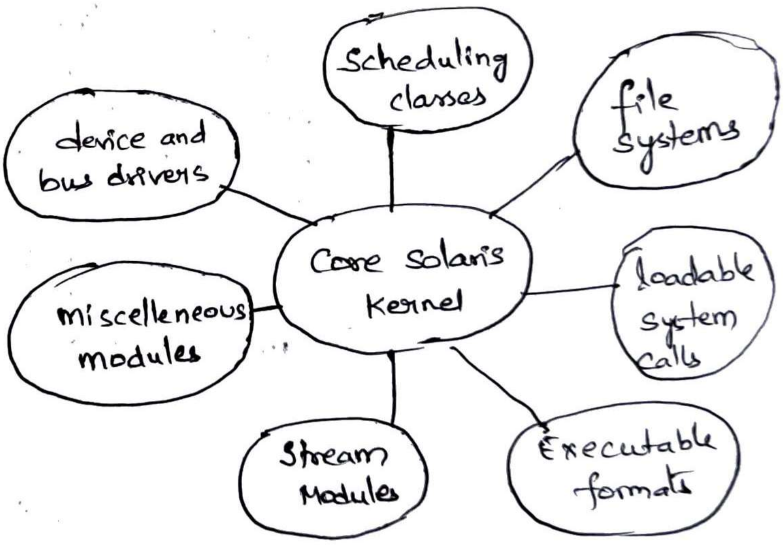


fig: solaris loadable modules