Syllabus:-Concepts and Generations of Operating systems, Services, Components, Types of Operating Systems, System Calls, Structure of an OS - Layered, Monolithic, Microkernel OS, Basic h/w support necessary for modern operating systems, Concept of Virtual Machine. Case study on UNIX and WINDOWS Operating System.

1.0 Introduction:

This chapter describes why OS is required, what is OS, functions of OS, etc. This chapter mainly focuses on evolution of OS on the basis of processing, user and special purpose OS. This chapter also focuses on views of OS i.e. user and system view. It elaborates the concept of system call, events and their types. In the middle of chapter, we will discuss the heart of OS i.e. kernel concept and their types. This chapter focuses on Hardware protection, Design and Implementation of the OS on the basis of different parameters. At the end this chapter we will focus on some case studies of different operating system like android, UNIX, windows etc.

1.1 Background of Operating System

The earliest computers were mainframe which does not have any form of operating system. Each user had a sole use of the machine for a scheduled period of time and would arrive to the computer with program and data. It often stored on punched cards, magnetic or paper tape. The program would be loaded into the machine and the machine would be then set to work with the program completed or crashed. Symbolic language, assemblers and compilers were developed for programmers to translate symbolic program code into machine code that previously would have been hand encoded. Later machine came with libraries of support code on punched cards or magnetic tape, which would be linked to the users program to assist in operations such as input and output (I/P and O/P). The true descendant of early operating system is what is now called the ***"Kernel".***

In technical and development circle the old restricted sense of an operating system persist because of continuous active development of embedded system (OS) for all kind of devices with a data processing components, from hand held gadgets up to industrial robots and real time control system, which do not run user application in the front end as a embedded operating system in a device as well. In today's era which is full of portability and flexibility, which will be very useful for all smart phones and probably in coming years it will be unique to all desk-top and mobile devices as well. The history of operating system has been tightly related to computer architecture, like generation of computer, functional unit, and memory management, processing activity and more things that we have already learned in computer architecture and organization subject.

1.2 Motivation of Operating System

Computer is an electronic device which performs arithmetic, logical operation and has data processing capacity. It provide some functions like data sharing, data processing, data moving, and data control (specially 4D concepts). Computer system is a combination of hard ware and software components. In Computer System the devices and users always wants to communicate with each other to perform dedicated task. These all connected devices are useless in absence of special program or system software, which can take responsibility to control operation of devices. Generally all devices are available with computer system has its own languages. Simply they all are different in nature and user doesn't understand their language. In order to understand the language of hardware there is requirement of system software. In fact operating system itself is system software which helps, in working & controlling of all such devices. It leads to development an Operating System. Everyone required (Hardware and User) an easy interaction to computer system. It consists of some processing elements as well, so computer is called as data processing unit. To synchronies the activities of different hardware and processing devices we would require an interface which will help to interact among them. One of the most important reasons to study OS is to learn how to extract the most efficient performance from it. The OS also provides a wide range of functions that assist in program execution.

1.3 Objective of Operating System

An OS is a program that controls the execution of application program and act as an interface between application and hardware.

The objectives of Operating System are as follows:

- To provide convenient way to use.
- To provide pleasant and effective user interface ***'Look at Feel'***.
- To hide details of hardware by creating abstraction of necessary
- To allocate resources to process operating system controls how process (active agent) may access resource (passive agent).
- To utilized resource effectively.
- Ability to evolve, related to new development and modification.

1.4 Computer Hardware Devices support by modern Operating System

An OS is intermediate to user and hardware of the computer it runs on. It extends the computer instruction set and manages its resources. To work within it user must know a great deal about how the hardware appears to the programmer.

a) Processors:

The brain of computer is the CPU (Central Processing Unit). It fetches the instructions from memory and executes it. The basic cycle of every CPU is, it fetches an instruction from memory, decoding is done to determine its type and operand, finally executes it. CPU has general register used to hold variable and temporary results. Most computers have special registers that are visible to programmer. One of this is program counter which contains the memory address of next instruction. Another register is PSW program status world; it contains condition code bits, which are set by comparison statement. Processors are available as single core, multicore, superscalar, multithreaded and many more.

Generally CPU comprises of ALU (Arithmetic Logic Unit) and CU (Control Unit), where ALU performs operations and CU control all activity by generating control signals. When CPU performs any operation it demand contents from main memory for this purpose CPU places the address of memory location on the bus then CPU fetches instruction and data finally it gets executed. For the same, CPU has two main phases one is instruction fetching and other is execution.

An ALU is a digital circuit used to perform operations, it is a fundamental processing unit and part of CPU. ALU is divided into AU (arithmetic Unit) and LU (Logic Unit) An CU is a component of CPU which directs the operation of processor and it also coordinate and control the all device related activities. It directs the operation by providing timing signals. CU is digital circuitry under CPU. Generally CU receive external instruction/command which it converts into sequence of control signals that will be sent to CPU then it assigned to particular devices for the operation. Control unit has two main type Hardwired Control units and Micro programmed control unit. Where Control functions implemented in hardware and Control functions implemented in software

b) Memory:

The second major component in all computers is the memory. Ideally a memory should be extremely fast, abundantly large and quite be cheap. The top layer consists of the register internal to CPU, next to that is a cache which is mainly controlled by hardware device. This has L1, L2 cache levels. Main memory comes next in the hierarchy; this is the work house of the memory system. Main memory usually called RAM, (Random Access Memory) some time it's called as core memory. Memory is discussed in chapter 3, where memory basics, management scheme, allocation strategies, virtual memory etc are covered.

Finally it comes to secondary memory which has plenty of capacity (generally it is in GB) to stores the data that usually used for backup purpose, some of the secondary memory includes hard disk, floppy disk, optical storage and solid state devices. Disk is a secondary storage, cheaper than the RAM, it consist of tracks and sectors on cylinders. It is discussed in chapter 2 where disk management, scheduling algorithm, etc are covered.

d) Tapes: Magnetic tape used for backup purpose which holds very large data set. It's read operation is sequential. This is traditional storage devices.

e) I/O Device: The CPU and memory are not only resources that the OS must manage. I/O devices also interact with the OS. This device consists of two part controller and device itself. The device driver is system software which controls all device activities.

f) Buses: Bus is a path way to transfer the information or control signals over a device. Data bus, control bus and Address bus are the main types. The main purpose of bus is communication between the devices because as we know computer system comprise of many devices which would not be functional without bus.

g) Interrupts & Exceptions: Interrupts are the signals generated by IO devices for processor to get attention towards them. Processor informs to OS about errors & changes in device Status. Processor uses polling mechanism to perform the same.

1.5 Definition of Operating System / Concepts of Operating System

As we all know that Computer System consists of many devices (i.e. I/P, O/P, processing elements (Software and Hardware)). They wants to interact with each other for processing purpose so that outcome can be produced, which require system software. System software is a program which consist of many important routines, subroutine, function and preprocesses which can easily helps to generates O/P for the given I/P. In some cases Operating System is a small program which consists of some specific instruction (also special purpose program). Standard definition of Operating System is an intermediate between Hardware and Users. Computer system also consists some of resources (CPU, Memory, I/O) and needs a proper management to produce an O/P with synchronized manner Operating System is a resource manager. Operating System acts as a "government" because it controls all the resources and their activity in system.

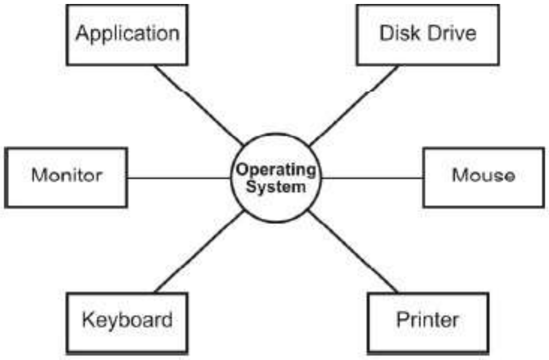


Fig.: 1.1 Overview of Computer System (OS).

Note: ***"Mother to Small Kids"*** Operating System is exact similar to this phrase consider ***Fig. 1.1*** which consist of number of devices are controlled by the single operating system. Technically Operating system is low level software that supports computer, basic functions such as task scheduling, controlling peripherals. Operating system is a collection of programs that control the application software which user run and acts as a link between the hardware and software running on the computer. Operating system is like conductor it is responsible for coordinating all individual computer components, so that they will work together according to single plan. Operating system is software which runs in kernel mode. Finally we can say the OS is an Extended Machine and Resource Manager. ***Ex: DOS, XP, Millennium (ME), windows server, vista, Linux, MAC operating system, Sun OS, Android, IOS, Blackberry, Symbian and many more.***

An operating system is defined as, a software component of a computer system and/or machine system that is responsible for managing various activities of the system as well as for sharing of system resources'.

1.6 Operating System View / Approaches

The operating system can be viewed as system perspective and user perspective. End user, application programmer, and operating system programmer use different aspects of a computer. The end user employs the application programs, the application programmer uses the system program/software to produce application program and the operating system programmer uses the hardware to implement system software.

User View:

The user view can be implemented as a top down approach. In a top-down implementation, it starts with the system call handler and checks what mechanism and data structure are needed to support them? These procedures are written until the hardware is reached. The problem with this approach is that it is hard to test anything with only the top-down procedure available. For this reason many developers find it more practical to build system bottom up. This approach hides the low-level hardware mechanism. Simply we can define a user view of system/OS is, in user view users uses the services provided by OS. User gets virtual machine to use, and it provided by OS itself. User need not to bother about how these activities were provide by the Operating System (i.e. need not to worry about background activity of OS). Simply user would be very happy to see "***Look at Feel***" i.e. what they see? This approach is also called top-down approach because user uses services as it is provided by the operating system.

System View:

Simply we can define a system view as, where OS acts as "***Resource Allocator***", program controller. Consider the situation in multi processing environment the OS effectively assign the resource to each process. OS is responsible to carry out all internal activities without knowing to the user. By considering all the system services it manages and controls all the activities. This approach is also called as ***"bottom-up approach***".

1.7 Services of Operating System/Component of Operating System

As we know operating system is service provider in computer system. Operating System provides many services that are usually categories into user and system perspective. Operating System services may vary from system to system.

1.7.1 User Perspective:

a) Program execution
b) Input / Output operation
c) File system manipulation
d) Process communication
e) User interface
f) Error detection

a. Program execution:

Program consists of set of instruction in passive form that needs to be executed by central processing element (CPU). For execution of any program the content of program should be loaded into main memory, after that execution will possible. For program the allocation and de-allocation of memory is done by operating system itself. Executable environment is provided by the operating system itself. The content of instruction is fetched from other location for execution purpose. For program execution straight line sequencing is required where program counter holds an address of next instruction that is to be executed. CPU has ALU, need for execution purpose by accepting valid inputs from user. More about this will discussed in chapter 3

b. Input / Output operation:

While execution of any program, many times, it requires some I/O devices for subsequent execution. (For example reads or write operation). All programs have their CPU burst and IO burst while execution program goes through both the burst for successful execution. After CPU burst there is an IO burst to any program. That I/O device (I/O burst) is provided by OS itself. User need not to control the I/O operation (allocation, release, use). Both the burst required context switch from CPU to IO for small amount of time.

c. File system manipulation:

File is a logical structure to store data on secondary storage where programs contents can be resides. User can write program contents in file and need not to worry about the content stored/write on secondary storage because Operating System keeps track of all such memory location, so that it will not get overlapped or override with other user data content. Hence user gets its own copy without any hazard. Apart from this, operating system provides creation, updating, and controlling privilege to file. Generally system have file system formats like FAT, FAT32, exFAT, NTFS etc. Each file system has its own formatting style. More about file operating will be discuss in chapter 2.

- Program needs to read a file or write a file.
- The operating system gives the permission to the program for operation on file.
- Permission varies from read-only, read-write, denied and so on.
- Operating System provides an interface to the user to create/delete files.
- Operating System provides an interface to the user to create/delete directories.
- Operating System provides an interface to create the backup of file system.

d. Process communication:

During the execution of program it required some value from other program, which requires (if co-operating process) interaction among/between both processes to produce the O/P. Such a process communication is provided by operating system through standard mechanism like IPC, socket, pipe, shared memory (will discusses it later) as a utility or mechanism which leads to proper communication among the processes. When co-operating process execute it affect other processes and also their output, depending on other process contents so proper synchronization is required. As mentioned above the standards provide some primitives to handle such activity, otherwise it wouldn’t have been any sequential execution.

e. User Interface:

As a definition of operating system (intermediately) it support or provide a way to user so that user can interact with hardware. OS provides graphical user interface (GUI), user interface (UI), command line interface (CLI), and batch processing to interact with user and hardware. With this mechanism user can interact with operating system by giving the commands and will get O/P for respective I/P.

A GUI (Graphic User Interface) is a graphical representation in which the users can interact with software or devices through graphical icons. It is very easy and interested way to interact. Whereas A CLI (Command Line Interface) is a console or text based representation in which the user types the commands to operate the software or devices.

f. Error detection:

Every process (during execution) is/has a sensitive issue because during execution the contents of program resides in main memory, and most of the time it transfer from one place to another place, which may lead to some hazard. Operating System keeps track of resources to monitor a normal operation or not. Operating system takes a corrective action after detection of an error and shows the error message to user so that user will know that error has occurred in system. Errors can occur anytime and anywhere. An error may occur in CPU, in I/O devices or in the memory hardware during operation. Following are the major activities of an operating system with respect to error handling.

- The OS constantly checks for possible errors.

- The OS takes an appropriate action to ensure correct and consistent computing.

1.7.2 Service to System Perspective:

a. Resource allocation
b. Accounting
c. Error Detection

a. Resource Allocation/Manager:

Computer system consists of many resources (CPU, Memory, I/O etc.). In multiprogramming system, many programs were executed at time. Program required some resources to perform their task, operating system itself allocate these resource. Operating System performs this activity with scheduling, timers and assigning the processes to the processors. The memory management carried out with memory management unit (MMU) to achieve proper allocation of memory to program through the base and limit register. Secondary storage management (HD, Magnetic Tape, CD ROM) is achieved by device drivers (system S/W) because each device is having different characteristics so they need proper synchronization through lock and release operation.

b. Accounting:

In multiuser system, more than one user performs their operation/task on computer system. Each user uses many resources many times as per requirement. This log must be maintain by the OS to take proper decision while allocation and reallocation of resources. Operating system keeps track of all users who logged in/into system, their usage of resource, their usage time and many more. Operating system keeps track of all users and resources present at system so it can take a proper decision in future. Accounting data is used for trend analysis, capacity planning, billing, auditing and cost allocation. (Dead lock, Memory management and Security).

c. Protection:

As per the system perspective, protection is a main concern. Protections of resources were done by operating system. Operating system provide a protection like deciding owner of file, ensure that resource are used logically and provides authentication, password management, locking mechanism, access matrix and many more provided by operating system. More about the same is explained in chapter 6.

Note: *The service of OS satisfies to both a user and system in fare manner.*

1.8 Evolution of Operating System

Brief History of Operating System:-The first computer didn't have OS, and could run only one program at a time. Early computer lacked from of OS the application talked directly to the hardware. In 1945 electronic computing circuits were introduced with separate functioning and didn't really need a programming language to be tested. In 1950, transistor made the computer circuit solid state, which eliminated the need to constantly patrol the innards of the computer to find burned out tubes and clean bugs out of the electrical construct. By 1965 the 1st OS like program were beginning to show up, like the Fortran maintaining system, which had to deal with loading three types of programs, the Fortran program, the compiler and machine language program. Along with the idea to run the card reader from one partition to another, then timesharing and many variation of processing, real time OS, clustered OS and many more evolved which we will discuss one by one. Apart from desk top system the OS has made tremendous achievements in mobile, embedded system as well. Categories of OS evaluation are based on processing, user, and application.

a. Sequential Processing:

In sequential processing the execution of task is performed in serial order i.e. one by one. It means at a time only one activity can be perform by processing element. Because of sequential nature of operations (single) it leads to time consumption. The processing element will have capability to handle one task only, and processing occurs in the order task is received. Programmer interact directly with computer hardware i.e. no operating system. Program in machine code is loaded with I/O device like card reader then operation is performed. After execution of one task user need to insert next task to processing element i.e. no automatic processing.

Alternatively referred as serial processing, sequential processing is a term used to describe the processing that occurs in the order that it is received. Sequential processing is in contrast to parallel processing or multitasking.

Drawbacks

- Setup time is high because, it is hardware dependent we need to do programming on hardware then it can perform specific task as per system program (driver) processing.
- Loading time.
- Less Resource utilization and time consuming activity.

b. Batch Processing OS:

As we have observed some drawback of sequential processing, and in order to overcome this problem, batch processing was introduced, where all processes are collected and then it submitted to the operator. Batch processing is a form of data processing in which a number of inputs are grouped. Then operator will sort all programs into batches in different categories. (i.e. compiler oriented like FORTRAN, PASCAL etc., as shown in **figure 1.2**) Actually the data is written on punch card and submitted to computer Operator then operator handles the task.

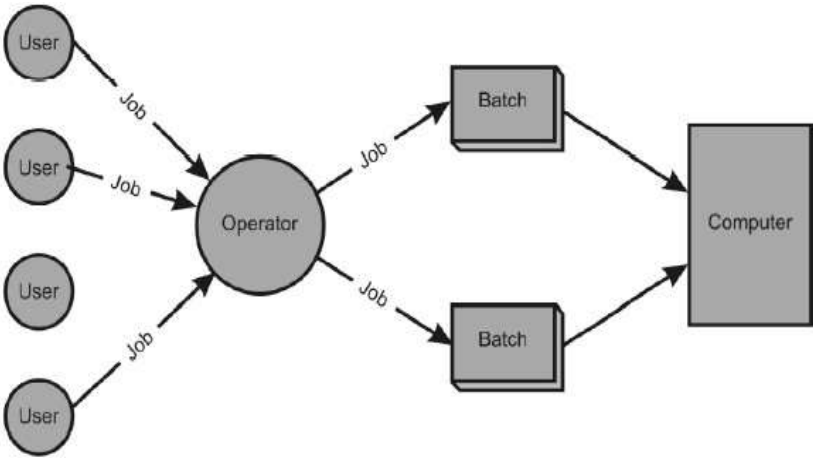


Fig1.2: Scenario for batch processing

Then the operator submits it to processing element and finally processor will perform entire processing.

Advantages:

- Batches are beneficial for large jobs.
- User can not interact with process execution.
- User can submit their program and take O/P after some time, so would beneficial to large size task.
- It increases CPU utilization by keeping processor busy.
- It is also possible to give/provide priority to processor.

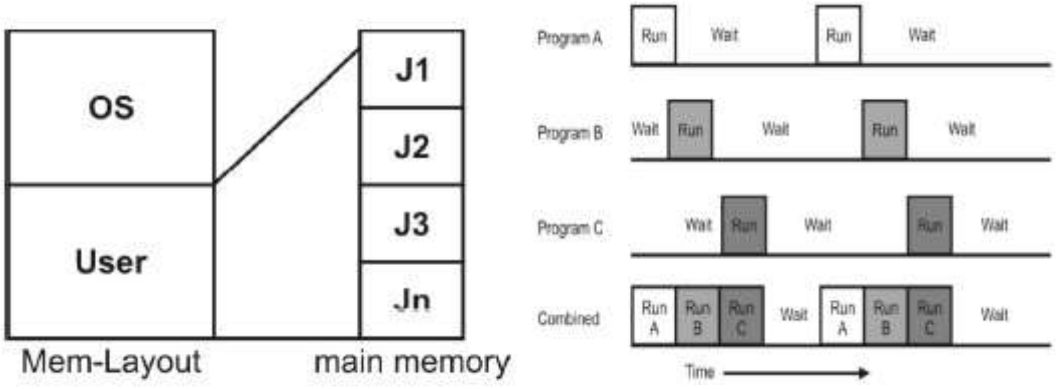
Disadvantages:

- If process is small in size it has to wait for other process.
- Time consuming to final execution.

E.g. - Bill processing, Pay roll system.

c. Multiprogramming:

To overcome the problem of under utilization of CPU and main memory, the multiprogramming was introduced. The multiprogramming is interleaved execution of multiple jobs by the same computer. In processing mechanism multiprogramming is one of the Processing where more than one programs can "***Resides in main memory***" at a same time. With the help of memory management technique memory is allocated to processes without any interleaving. When program resides in main memory user can not interact with system all the decisions are taken by system itself. (user can not have choice to select process). One system module will select proper process and submit it to processor (From main memory) for example consider **figure 1.3**(case a) where memory layout is divided into OS and user control. The user control space consists of four task (J1, J2, J3, and Jn) that means at a time all 4 task are present (resides) in main memory. Now it's a duty of system (scheduler) to select one of process and give it to the processor. Generally all processes are having CPU and I/O burst during their life time, it frequently switches from one burst to another burst during operation.



Case (a) Case (b)

Fig.1.3: Multiprogramming with three programs

The multiprogramming is possible due to such activity where one process executes their CPU burst and another can busy with I/O activity to achieve multiprogramming. It happened just because of context Switching mechanism (Case b). Consider general example of word and excel application both are opened. If we are typing in word application excel just present in main memory. The main memory content has not swapped out yet because it might be possible to start its operation subsequently. Another example is if users write some code in different languages but only one can active at a time while others are only resides in memory. The remaining processes were waiting for CPU turn so that they can start execution. Multiprogramming takes the help of memory management technique like paging and virtual memory to allocate the space properly. Note that if there are N ready processes and all of those are highly CPU-bound (i.e., they mostly execute CPU tasks and none or very few I/O operations), in the very worst case one program might wait all the other N-1 ones to complete before executing.

Passive and Active multiprogramming concept also are available, although the two terms are used interchangeably nowadays, the difference lies on the design of the programs that triggers the processes. How frequent it can be interrupted to give way to another process. In multiprogramming a list of programs are queued and switch programs from active to passive states. If the program turns into a passive state another program would be considered as active.

Whereas in multi-tasking, it tries to pause/stop the current running process instantly regardless of the state and executes another process instantly. When a program that is being processed by the CPU issues an I/O command, control is passed to the I/O processor and the CPU is free to process other programs. This is known as **passive multiprogramming**. Passive multiprogramming is characterized by an event which generates interrupt that cause task switching. These events are usually the result of a program-initiated action, such as sending a record to an output device, or dividing by zero. The problem with this is that the interruptions are primarily controlled by the programs and not by the CPU. As an answer to this problem, the CPU was given the control with the advent of **active multiprogramming**. Active multiprogramming enabled a program to use a preset slice of CPU time before putting the program on hold to process another program – Active multiprogramming is characterized by time generated interrupts that cause task switching. The time interrupt is generated by a system clock monitored by the operating system

Note: *The number of processes that the system supports is called the degree of multiprogramming.*

d. Multi Tasking OS:

Multitasking is an ability to do more than one task at a time (but practically it is not possible). Earlier system was single tasking (MS-DOS). But after decade researches have been made and it leads to the invention of multitasking system. (WindowsNT) Concept of context switching is required to switch execution from one process to another process. E.g. if a man is listening song at the same time he or she can browse the content from web. In order to achieve multitasking we need to share some shared resource (CPU and Memory). There are two basic types of multitasking system i.e. preemptive and co-operative. In preemptive multitask, the OS distribute the part of CPU time slice to each program and in co-operative each program can control the CPU for as long as it needs. In multiprogramming and multiuser system user can interact with system i.e. it can control the execution of multiple program. – E.g. Reading a news paper and taking sip of tea. In multitasking (modern OS) time sharing is best manifested, because each running process takes only a fair quantum of the CPU time.

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

e. Multiprocessing:

Multiprocessing is also called multicore programming. It works on the principle where large task is divided into smaller sub task and delegating those task to more than one processor which are connected (tightly) to computer system (e.g. parallel processing). Simply we can say that multiprocessing is a mechanism in which two or more program or segment of instruction simultaneously run by computer (CPU). In multiprocessing the processor can be used to execute a single sequence of instruction in multiple contents (single instruction multiple data SIMD) often used in vector processing. This processing is of two types, one is the multiprocessing which involves computer H/W and S/ W architecture where two or more identical processor are connected to a single shared main memory. That called highly compiled system and symmetric multiprocessing (SMP) it does not carry master slave relationship between processor (**Fig.1.4**). It also has its own set of register and caches (All processor). Where as in an asymmetric multiprocessing the system is loosely coupled in which all CPU are treated equally (**Fig.1.5**). For example system might not allow (either at the H/W & OS Level) one CPU to execute OS code or might not allow one CPU to perform I/O Operation, Such environment exist master slave relationship among processors. Server having rights to assign task to same processor but individual processor may perform task on its own processing element and works as uniprocessor system.

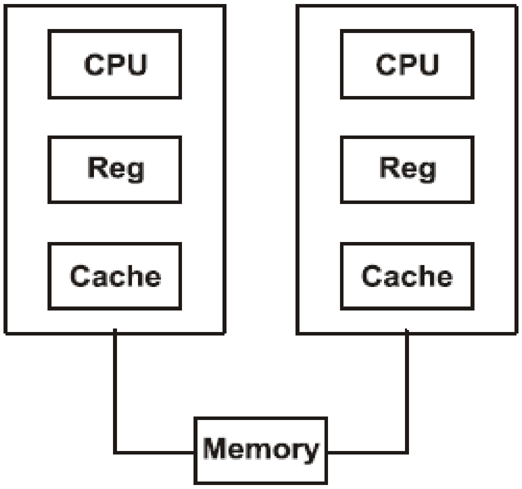


Fig.1.4: Symmetric M-Processing

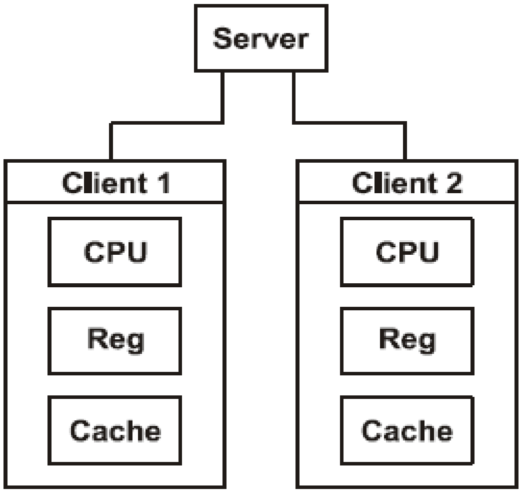
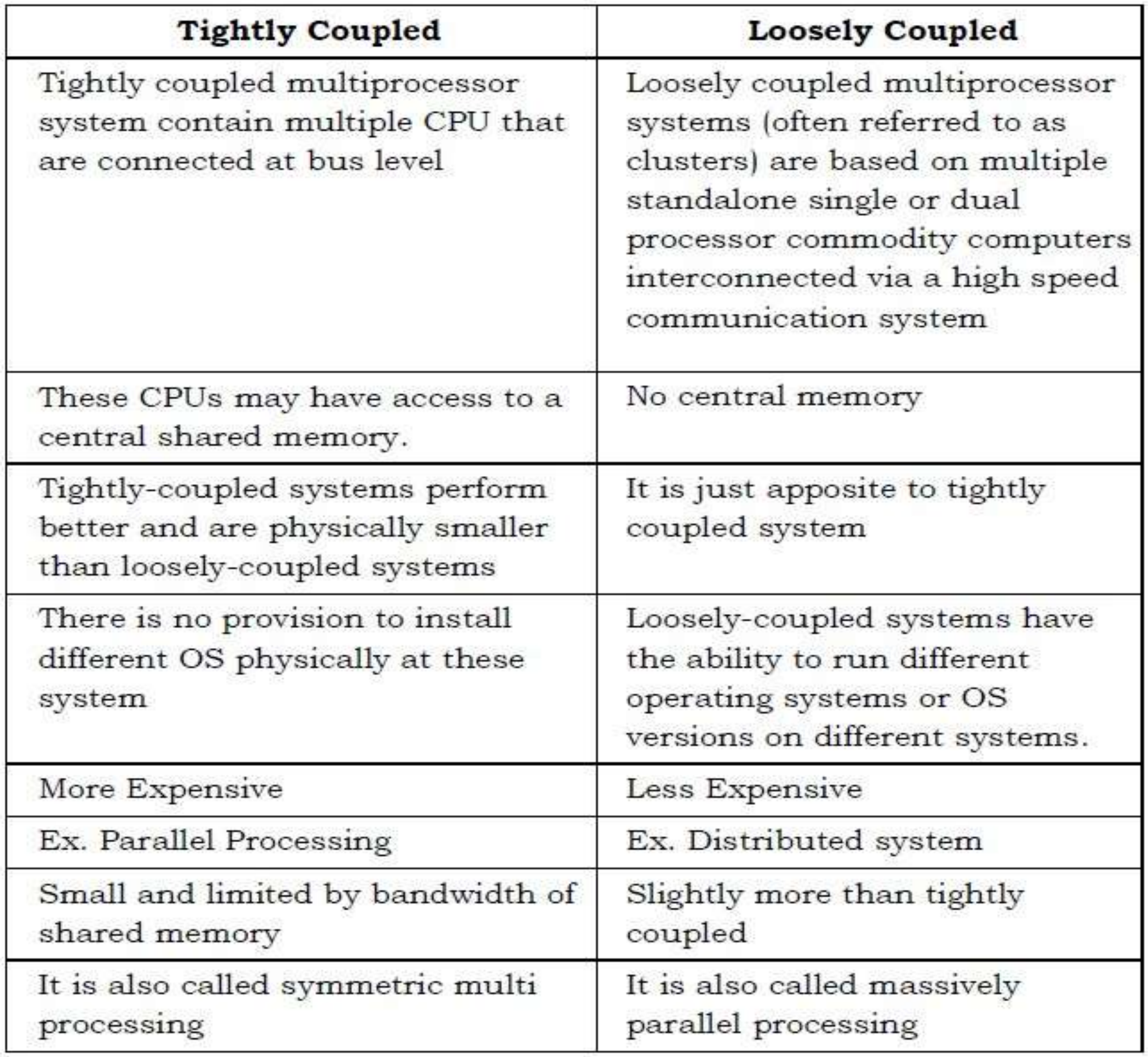


Fig.1.5: Asymmetric M-Processing

Note:-In multiprocessing tightly coupled and loosely coupled are two concepts



Tightly Coupled	Loosely Coupled
Tightly coupled multiprocessor system contain multiple CPU that are connected at bus level	Loosely coupled multiprocessor systems (often referred to as clusters) are based on multiple standalone single or dual processor commodity computers interconnected via a high speed communication system
These CPUs may have access to a central shared memory.	No central memory
Tightly-coupled systems perform better and are physically smaller than loosely-coupled systems	It is just apposite to tightly coupled system
There is no provision to install different OS physically at these system	Loosely-coupled systems have the ability to run different operating systems or OS versions on different systems.
More Expensive	Less Expensive
Ex. Parallel Processing	Ex. Distributed system
Small and limited by bandwidth of shared memory	Slightly more than tightly coupled
It is also called symmetric multi processing	It is also called massively parallel processing

f. Single User OS:

Operating system evolution is also categories on the basis of how many users can use a system at a time (i.e. single user or multiuser at a time). Single user OS is convenient to the single user to perform dedicated task. In single user OS, other user can not interact with working users (at a time only one). The core part of single user OS is one kernel image that will run at a time i.e. there is no other facility to run more than one kernel images while booting e.g. DOS, Windows 95 (most of UNIX version are single user mode). Nowadays, it is not too convenient because everyone need multiuser interaction and large processing. Single user OS is commonly used for home computer system. These types of OS are typically used in small devices like wireless phone and two way messaging devices. It is not useful for a computer or other device, intended to run multiple programs at once.

g. Multiuser OS:

Multiuser OS is an application or S/W that allows multiple users to access any system. The concept of Multiuser OS is as; if any OS having multiple kernel image is loaded while the user login to the client computer. (Like a thin client environment). When user boot their system then single kernel image is loaded for specific user. In simple term, multiuser OS are those in which multiple kernel images get loaded with number of users. It is good provision for all users to complete their task by simply remote login or login to server system. The nature of S/W or application is inbuilt multiuser (most of the time). Consider, an application as an antivirus which is having multiuser functionality i.e. it having a capacity to handle more user at a time. Every user will get separate environment for processing. In thin client environment, when dump (diskless) computer are logged in then single kernel (OS) copy is invoked from server e.g. UNIX, Linux, and Mainframes such as IBM AS400

h. Time Sharing OS:

Time sharing OS is a logical extension of multiprogramming. It means it is possible to have more than one processes in main memory at a time. (i.e. Time Quantum) **Figure 1.6** shows the detailed view of time sharing OS.

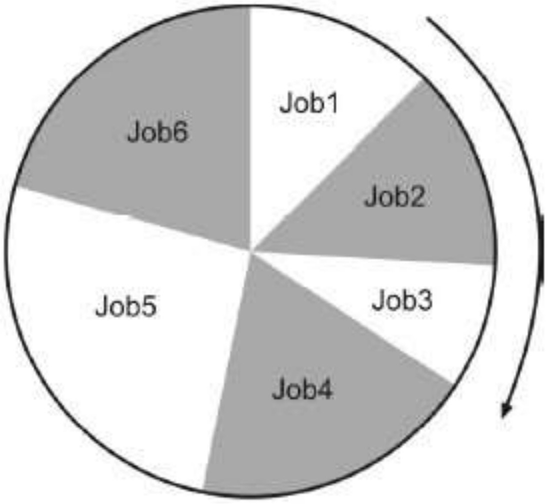


Fig.1.6: Time sharing OS

If a process completes their execution within a given time slice then no problem but if any process remains partially to execute their portion then it will get next time quantum for execution purpose after other process completes. In order, to provide more interactiveness and save response time it is most preferable. In time sharing, shuffling of job is possible on time quantum basis. It is an illusion of several job executing simultaneously. The use of scheduling and context switching are needed. In time sharing system resources are shared among different jobs for their execution.

Advantages:

- Time sharing system reduces CPU idle time.
- It leads to quick response.
- It is convenient to small user.
- Effective use of resource by sharing to different processor.

Disadvantages:

Time sharing system leads to some drawback due to the nature of interactiveness, multiprocessing and multiuser operation.

1. Question of Security:

In time sharing system is arises because of the nature of multi-user and interactiveness of time sharing. Here, more than one user can run their code at a time on same platform therefore there may be a possibility of having interleaved of data contents or the program values. And hence the contents of process can be accessed by any other process.

2. Problem of Communication:

In time sharing operating system, the execution of processes switches from one process to another process as per time quantum. At the same time it needs to save the current status of process and need to reload their content frequently that are required for process communication (IPC). This may cause of interleaving of memory address space and program content. Shared memory communication leads to keep more track of each process, and hence communication will get difficult.

3. Question of Reliability:

The interactive nature of Time sharing system causes unreliability because the data content may be lost which will lead to failure.

i. Parallel OS

Parallel processing is a superset of multi processing system. Parallel processing works on the principle, where the large task is divided into small sub task and is assigned to individual core present in computer system and finally all O/P will merge together. Consider scenario in **figure 1.7** where a large task is divided into three subtasks then they are assigned to individual core and the result gets merge together.

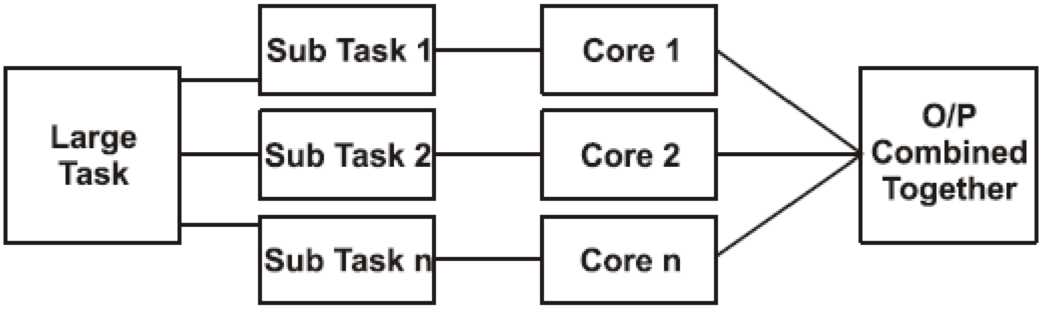


Fig.1.7: Parallel Processing

Parallel processing is tightly coupled (as discussed in multiprocessor) system in which the resources are shared by all processing elements (like memory, clock). In order to communicate with different processes it requires inter process communication (IPC), and it is achieved by shared memory mechanism. Data dependency concept is important if subtask1 depends on subtask 2, so they have to wait for each other, to produce an O/P. Dependency may be in the form of I/P dependency, O/P dependency, flow dependency, etc.

Advantages:

1. Throughput increases

Presence of multicore architecture leads to more outcomes. The speed up ratio with N processor is not N; rather it is less than N. When multiple processor co-operate on a task a certain amount of overhead is incurred in keeping all the parts working correctly. Simply, the availability of more processing elements leads to perform more tasks within less time so that maximum throughput can be achieved.

2. Economic of Scale

If we compare a single processing and multiprocessing environment it saves lot of resource (memory). For example, if we purchase single system it requires all components to produce output but if we shared those resources it would save some money. Even, if several programs operate on the same set of data it is cheaper to store those data on the disk and to have all the processor share them than to have many local disks.

3. Increase Reliability

Because of availability of many replicated resource, there is fault tolerance. If one of the processing elements gets failed it does affect other processing element. Hence user can simply rely on multiprocessing system.

4. Last advantage of parallel processing, is that it works on divide and conquer rule in which lot of advantages are possible.

j. Desktop Operating System

The desktop OS is particularly used in personal system. Desktop OS is a special purpose operating system. In the 1st generation of computer, the processors were lacked the feature need to protect an OS from user programs, therefore it is neither multiuser nor multitasking. But the goal of this OS changes with time, instead of maximizing CPU and peripheral utilization; it is used to maximize user convenience and responsiveness. Even the cost is sufficiently less than other system. Adoption is main feature of desktop OS; it can support maximum user level S/W installation. It is also possible to run more than one OS and S/W; it is more interactive and responsive.

k. Network Operating System

As compared to desktop OS the NOS (Network OS) has separate system images for every user. NOS’s are the middleware between local OS of one computer to local OS of another computer. Many computers are connected via some connecting device which ultimately forms a network. It’s a responsibility of sender and receiver that, transfer takes place properly. During this time, the total care of task at network level is cared by NOS along with network paradigm and format. NOS are a loosely coupled OS where each system has their own components (processing element). During data transfer on N/W the concept of marshaling and demarshaling (encapsulation) is needed i.e. only end user can access data while middle one can't. The main (primary) task of NOS is sharing of resource. NOS is the extension of traditional OS and it is specially used for Resource sharing and information sharing on the networks.

Network Operating System is an operating system that includes special functions for connecting computers and devices into a local-area network (LAN) or Inter-network. Some popular network operating systems are Novell Netware, Windows NT/2000, Linux, Sun Solaris, UNIX, and IBM OS/2. NOS use high level API, Socket, RPC for process communication. NOS are less fault tolerant than distributed OS

Applications:

- Remote Login
- File Transfer
- N/W Browsing
- Remote Execution
- Messaging

l. Distributed OS (DOS)

Distributed system is a special purpose OS (Advance OS). Distributed OS is an extension/advancement of Network OS with higher level of communication. DOS is a comparatively more fault tolerance system than NOS. DOS is also a loosely coupled system where each computer has its own computational/processing elements/resource (CPU, Memory, etc.) DOS can be define as a collection of autonomous computer systems which are capable of communication and cooperation with each other's through high capacity LAN/WAN network. It can also define a collection of independent computer that appears to its user as a single coherent system. Transparency is a key design issue of DOS where every details are hidden from user, user can’t get actual status of activity, like replication, Resource, N/W, etc. Common user cannot predict the execution place (location) on distributed environment. As in distributed environment all the workstations/systems are geographically separated.

Purpose of DOS:

- How to manage resource effectively
- Transparency issues
- Replication facility
- Large and effective data processing
- DOS lacks in common shared memory and clock (global) which leads to many problem like event ordering, synchronization, etc.

Advantages:

- Resource sharing
- Computational speed up
- Reliability
- Scalability

E.g. Mach OS, Chorus OS, etc.

Parallel vs Distributed Computing	
Parallel computing is a computation type in which multiple processors execute multiple tasks simultaneously.	Distributed computing is a computation type in which networked computers communicate and coordinate the work through message passing to achieve a common goal.
Number of Computers Required	
Parallel computing occurs on one	Distributed computing occurs between multiple

computer.	computers.
Processing Mechanism	
In parallel computing multiple processors perform processing.	In distributed computing, computers rely on message passing.
Synchronization	
All processors share a single master clock for synchronization.	There is no global clock in distributed computing, it uses synchronization algorithms.
Memory	
In Parallel computing, computers can have shared memory or distributed memory.	In Distributed computing, each computer has their own memory.
Usage	
Parallel computing is used to increase performance and for scientific computing.	Distributed computing is used to share resources and to increase scalability.

NETWORK OPERATING SYSTEM	DISTRIBUTED OPERATING SYSTEM
Network Operating System's main objective is to provide the local services to remote client.	Distributed Operating System's main objective is to manage the hardware resources.
In Network Operating System, Communication takes place on the basis of files.	In Distributed Operating System, Communication takes place on the basis of messages and share memory.
Network Operating System is more scalable than Distributed Operating System.	Distributed Operating System is less scalable than Network Operating System.
In Network Operating System, fault tolerance is less.	While in Distributed Operating System, fault tolerance is high.
Rate of autonomy in Network Operating System is high.	While The rate of autonomy in Distributed Operating System is less.
Ease of implementation in Network Operating System is also high.	While in Distributed Operating System Ease of implementation is less.
In Network Operating System, All nodes can have different operating system.	While in Distributed Operating System, All nodes have same operating system.

m. Real Time OS

A key characteristic of RTOS (Real Time OS) is the level of its consistency concerning about the amount of time it takes to accept and complete the task. RTOS is a one type of multiuser OS.RTOS always givens O/P in definite time. RTOS is specially used for real time (server) application. It has ability to process data without buffering or delay. The time constrains is a main issues in RTOS. e.g. Airplane (Landing, takeoff), lift, radar, airlines reservation system, air traffic control system etc.

Real time Operating Systems are very fast and quick respondent systems. These systems are used in an environment where a large number of events (generally external) must be accepted and processed in a short time. Real time processing requires quick transaction. And it is characterized by supplying immediate response. Job scheduling is the main concern of RTOS, it need to be materialized in proper sequence.

Types of RTOS

1. Hard RTOS

- It produces O/P in definite time
- If O/P is not produced within the time it leads to huge loss of data.
- It has less delay than soft RTOS.
- It needs advance algorithm for job scheduling

2. Soft RTOS

- It is having less time constraints as compared to hard RTOS
- If O/P is not produced within the time then loss is considerably less than hard RTOS.
- It has greater delay than hard RTOS.

Note:

Generally the sensors are used to take a data (live) from different places because RTOS believes/works on Real time data. RTOS need to take instance decision for that if required current data. It cannot tolerate any time delay. Job scheduling is needful to achieve real time activity.

Difference between in GPOS and RTOS

General-Purpose Operating System (GPOS)	Real-Time Operating System (RTOS)
It has a predictable behavior.	There is no predictability.
It works under worst case assumptions.	It optimizes for the average case.
It does not have a large memory.	It has a large memory.
It used for desktop pc, laptop.	It applied for the embedded application.
Process-based Scheduling used.	Time-based scheduling used like round robin.
Interrupt latency is not considered as much crucial as in RTOS.	Interrupt lag is minimal, measured in few microseconds.
No priority inversion mechanism is present in the system.	Priority inversion mechanism is current. Once priority set by the programmer, it can’t be changed by the system itself.
Kernel operations may or may not be preempted.	Kernel operation can be preempted.

N-Cluster operating system

Cluster OS is a type of multiple CPU system which can accomplish the high speed computational works. Cluster system has more CPU but at different location and they communicate with a connected networks like LAN, MAN. Clustering is usually used to increase availability of services. Cluster may be of same type or different types i.e. homogeneous or heterogeneous. Cluster OS is a combination of networks OS.

Clustering technology enable multiple servers to work in unison to create the appearance of single computing environment. Technically, each server is running its own operating system but they work together as if they are one. Some cluster technology example are:- Service guard cluster, Extended campus , Metropolitan cluster ,Continents cluster.

O-Client server operating system:-

Now a day's computers have become much faster, more powerful, and cheaper. The designer have shifted from traditional centralized system architecture to a new paradigm because centralize model have drawbacks. Where a user terminal gets connected to centralize system and is supplemented by computers. As a result today's systems, acts as a server system model to satisfy the request generated by clients. The working of client server is simple. This system provides an interface which helps the clients to give the commands to the server. Basically, a client sends a request for server for action in response it returns executed part of process. This operating system works on the concepts of request–response model. **E.X.** servers, file server, mail server, time server, and many more.

P-Web-based operating system:-

Today's era is fully web based. To perform real time activity, each user believes on web because user can access their documents from anywhere anytime. Web computing has increased the emphasis on networking device that were not previously networked. now it includes wired and wireless network as well. Wired network is faster than wireless but it cannot support mobility, but wireless is more powerful. Normal users could not pay the direct amount for computation of high speed operation so it is better to take it on rent and use from house via any web server or service providers. The implementation of web based computing has given rise to new categories of devices such as load balancer. It distributes network connections among pool of similar server to manage performance. Operating system (normal) supports for web based activity. Generally, web has increased the complexity of device, because their users require them to be web enabled. Web based OS, are meta-computing terms that refer to network services for internet scale like distributed system. In simple words, a web OS is used to access web services via internet.

1.9 Kernel Concept

The OS is a complicated piece of software, which consists of a number of routines, subroutine. Sometimes it is a complicated piece of S/W. Obviously the size of the OS is very large and it is hard to keep the whole OS in memory all the time, as very little space would be left for other application programs, due to limited size of the memory. Therefore, the OS routine is divided into two parts. One part consists of very essential routines which are required very often and almost all the time, consists of routines which are required sometimes, but not always.

In this sense, they are not vital. The important portion is called as kernel of the OS. This is the inner most layers of the OS close to the hardware and control the actual hardware. It is the heart of OS. All the other routines are loaded from the disk to the memory as and when needed. **figure 1.8** depicts kernel of OS and **figure 1.9** depicts location of kernel.

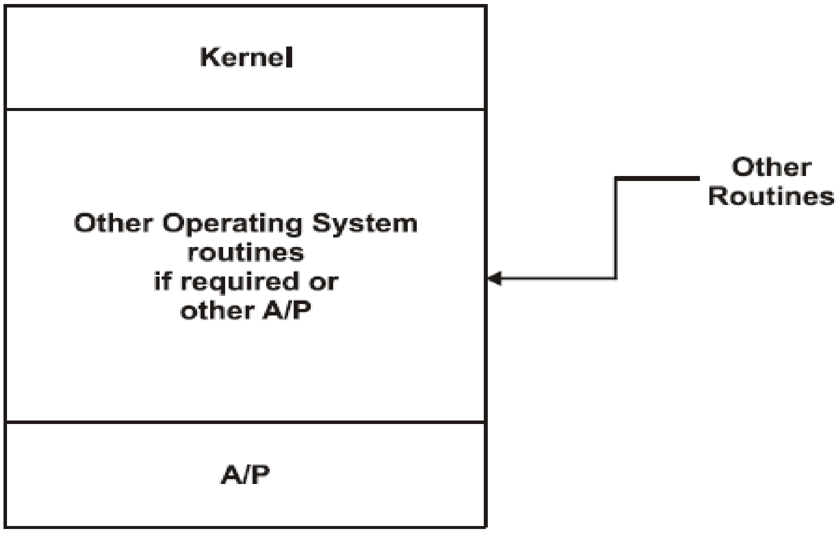


Fig.1.8: Kernel of the Operating System

Main Points:

- Kernel is a part of OS that loads into memory during boot time and remains present into memory as long as computer is switched off.
- Generally, kernel resides into protected memory area.
- If Computer crashes it means kernel is also crash.
- We can also say the Kernel is central module of computer system.
- Kernel is a place where the real work is done.
- Sometimes Kernel is called an Operating System as well.

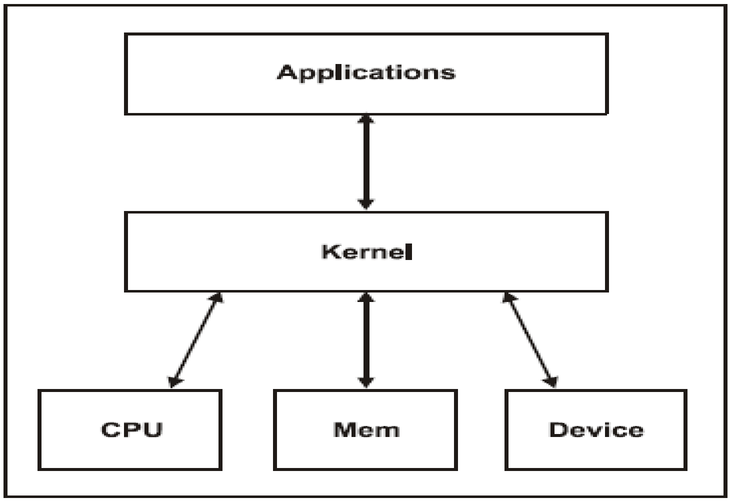


Fig.1.9 Kernel Location

1.9.1 Types of Kernel

There are two main types of Kernel is OS

A. Monolithic Kernel

B. Microkernel

A. Monolithic Kernel

Earlier in monolithic kernel architecture, all the OS services (process, memory management, interrupt handling, etc.) were packed into single module in kernel space. This leads to some drawback

1. Size of Kernel may become very huge.

2. Poor maintainability, which means bug fixing or addition of new feature requested in recompilation of the whole kernel which can consume lot of time.

Nowadays, modern approach for monolithic Kernel architecture is used, kernel consists of different modules which can be dynamically loaded and unloaded at run time.

Description:

Figure1.10 shows the memory layout of OS where user space and kernel space are shown. The entire OS is loaded into kernel space and it acts as a supervisor. All services are provided by the OS that is equivalent to monolithic kernel. If kernel fails then we would say entire OS has failed. The device drivers are loaded into the running Kernel and it become the part of available kernel. The function of OS involves simple function calls within the Kernel which is one layer program.

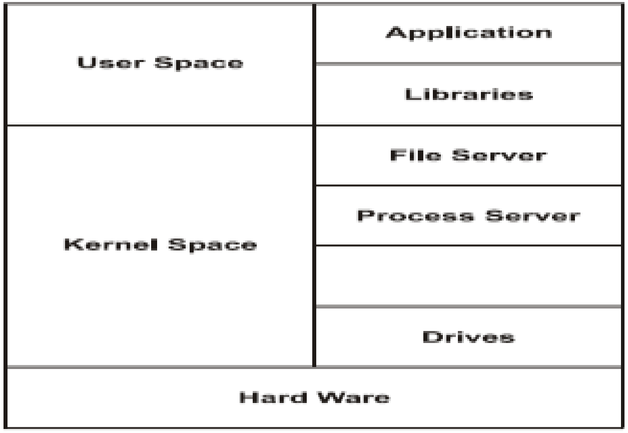


Fig.1.10 Monolithic Kernel

Kernel image = (kernel core + kernel services)

Monolithic kernel is nothing but entire OS which provides all services as OS.

B. Micro Kernel:

As we have seen in monolithic kernel, it requires abundant space to load kernel into main memory and it also leads to other problems. In micro kernel, all nonessential programs are kept in a user memory space and essential system programs are kept in kernel space. The communication between all components is provided by the message passing (IPC). In micro kernel, user has flexibility to extend system at any level. In microkernel, the kernel part is moved into user space.

Advantages:

- Extension of OS much easier.
- Kernel becomes smaller due to separation of services at separate layers.
- So changing, modification, maintenance and debugging is easy.
- It is more secure and reliable.
- It is also easy to port.

Disadvantages:

- Poor performance due to increased system message passing, content switching and other.
- It creates more confusion to other users because the code designed by one user may not be essential for other so they may get confused.

Kernel image = Kernel Core

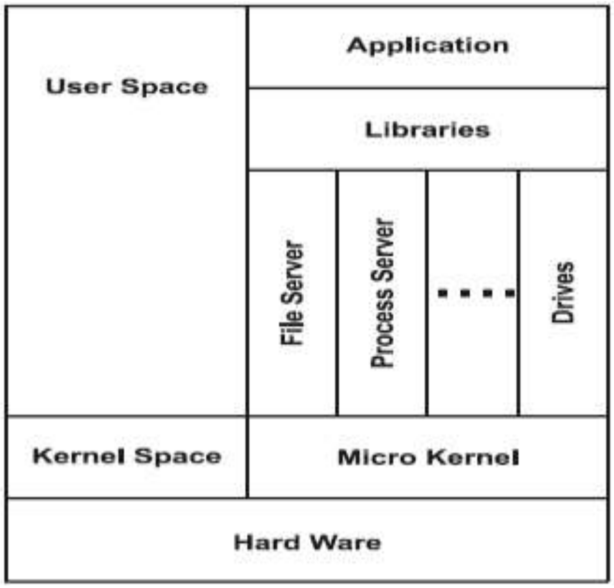


Fig.1.11: Micro Kernel

The microkernel can be considered compact kernel as it performs only the basic function universal to all computers. In **figure 1.11** the OS services are deployed at user space and only required code is kept at kernel space. When the special routine is required at that time it can invoke to kernel space.

Note: *There are also some other kernel types available like Hybrid, Nano and Exo.*

1.9.2 Kernel Entry Points

Event - In computing environment, event is an action or occurrences detected by the program that may be handled by the program. Typically, events are handled synchronously with the program execution flow i.e. the program has one or more dedicated places where event has been handled. The source of event comes from K/B or there I/O devices or any programming action which generate/focus the attention of CPU (Processing element). Event can also use instruction set. Simple definition of event is an action performed by user or program called event. An event object is used to call the events and event handlers are used to handle the events.

Types of Event:

1. Interrupt

Interrupt is an event generated by I/O device to take attention of CPU for execution purpose. When I/O device get hold of CPU that time the previous process contents saved into ISR (Interrupt Service Routine) where the actual location is presents. We have studied a lot about interrupt in Computer Organization & Architecture and Microprocessor subject i.e. type of interrupt (maskable, non-maskable, software and hardware interrupt).

Our keyboard is a very simple input device; simple because it generates small amounts of data very slowly (by a computer's standards). When we press or release a key, that event is signaled up the keyboard cable to raise a hardware interrupt.

It's the operating system's job to watch for such interrupts. For each possible kind of interrupt, there will be an interrupt handler, a part of the operating system that store away any data associated with them (like our key press/key release value) until it can be processed. What the interrupt handler for our keyboard actually does is post the key value into a system area near the bottom of memory. There, it will be available for inspection when the operating system passes control to whichever program is currently supposed to be reading from the keyboard. Every kind of interrupt has an associated priority level. Lower-priority interrupts (like keyboard events) have to wait on higher-priority interrupts (like clock ticks or disk events). UNIX is designed to give high priority to the kinds of events that need to be processed rapidly in order to keep the machine's response smooth

2. Trap

Trap is also one type of event which usually generates by the CPU itself. CPU analyzes the processing instruction whether the instruction is execution is really fruitful or not, if not CPU generates trap and control goes to operating system, e.g., Divided by zero operation, there is no sense to execute this instruction that time CPU takes an action (i.e. trap) to avoid unnecessary action and control goes to OS then OS display proper message to user that error has been occurred. Because CPU knew that divide by zero instruction is cause to waste of CPU cycle. The process contents need not to save on ISR because there is no as such requirement of these process states in future.

3. Signal

Signal is also one type of event/interrupt send to process, which is type of soft interrupt to process. If user want to stop the activity of any process (foreground) that time by using s/w interrupt they can control the process execution, e.g. while execution of any process the user can perform signal operation using keyboard like (Ctrl+C, Ctrl+Z,in UNIX and many more) and can control the process for a while. When signal raised/generates that time current activity will be interrupted and signal Handler can call to handle the signal event.

In case of signal the process may takes three actions.

1. It takes default action defined by the O.S.
2. It can ignore signal.
3. Or, last it does specific action (programmer).

Note: *Generally on background processes signal (soft interrupt) cannot perform any action means it does not affect or stop the execution of process.*

1.10 System Call

As we know that for performing any Operation as user must have to specify the Operation which he or she wants to operate on the Computer. We can say that For Performing any Operation a user must have to request for a service from the System. For Making any request a user will prepare a special call which is also known as the System Call.

System calls are interface provided to communicate with the O.S for the user. An OS system manages entire functioning of a computer by its own, but on many occasion explicit direct or indirect calls are required to perform various operations. Routines, functions or calls that are used to perform OS functions are called as system calls. Most of the system call of OS are available is the form of commands. System call instruction are normally available is assembly language. High level language such as C, C++ and perl also provides facility of system programming.

The System Call is the Request for running any Program and for performing any Operation on the System. When a user First Time Starts the System then the System is in the user Mode and When he request For a Service then the User Mode will be Converted into the Kernel Mode Which just Listen the Request of the user and Process the Request and Display the Results those are Produced after the Processing.

Why System call are required

- User process must not be given open access to the kernel code.
- The system call interfere layer contains entry point in the Kernel code.
- Any user or application request that involves access to any system resource must be handled by the kernel code.
- When program needs to access restricted resources.

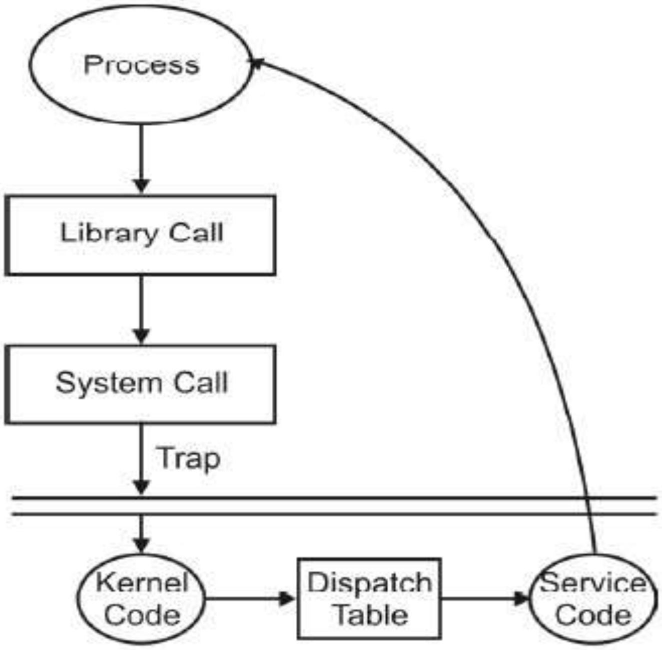


Fig.1.12: System call processing

Working: When user program wants to take services from OS e.g., Read, system, call. User program makes a call to the library functions. Then Library routine puts appropriate parameter at a well known place (e.g., Register, stack, Table in main memory). It passes library parameter or system call number. The trap instruction is executed to change mode from user to kernel. After that control goes to O.S. Then OS determine which system call's is to be carried out (by system call no.). Then kernel index to dispatch table which contents pointer to service routine for system call. After that service routine is executed and return parameter or error code are place at well known place usually a CPU register. At the last control goes to user process. A library function executes the instruction followed by the trap. (**Figure 1.12** shows system call working)

1.10.1 Types of System call

System call can be grouped into some categories.

(A) Process control system call.

- Creation of processes, Termination of process
- Abort of process, End of process.
- Load of process, execution of process
- Get process attributes, set process attribute
- Wait for time, wait event, signal event.
- Allocation and De-allocation of memory.

(B) File management system call

- Creation of File, deletion of file.
- Open, close, Read, Write and reposition of File.
- Get File attributes, set file attributes.

(C) Device management

- Request to specific device, used of device and Release.
- Read, Write, Reposition.
- Get/Set device attribute.
- Logical attach or detach devices.

(E) Information maintenance

- Get/Set time or date.
- Get/Set system date.
- Get/Set process, File or device Attributes.

(F) Communication

- Create, delete, communication connections.
- Send, receive, messages.
- Transfer states information.
- Attach or detach remote device.

(G) Protection

- Set file security.
- Initialize security descriptor.
- Set security, Descriptor Group etc.

1.11 H/W protection in OS

1. Dual-mode operation

Sharing system resource requires OS to ensure that an incorrect program cannot cause other program to execute incorrectly. In order to support protection, two modes of operation are provided, (**Figure 1.13**) namely user mode and system mode.

- In User mode the program execution is done on behalf of a user.
- In System mode it also called supervisor mode, here execution is done on behalf of OS itself. It allows those programs that are not allowed in user mode.

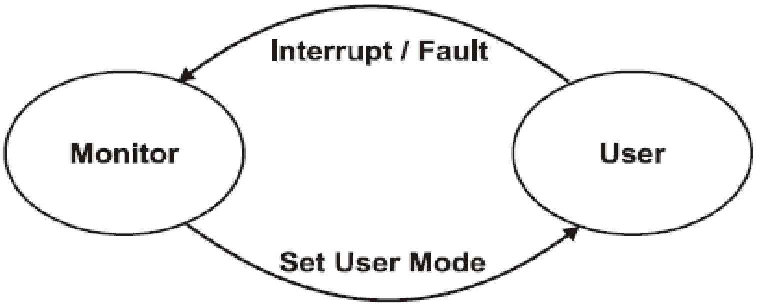


Fig.1.13: General Scenario

Solution: The mode bit is added in computer H/W to indicate the current mode i.e., system=1 or user mode=0. Privileged instruction can be issued only in monitor mode.

2. I/O protection

All I/O instructions are privileged instruction; they must ensure that a user program could never gain control of the computer in monitor mode (i.e. user program that apart from its execution, stores new addresses in the interrupt vector table). If user wants to take privilege, it needs to switch with the help of System call. (**Figure 1.14**)

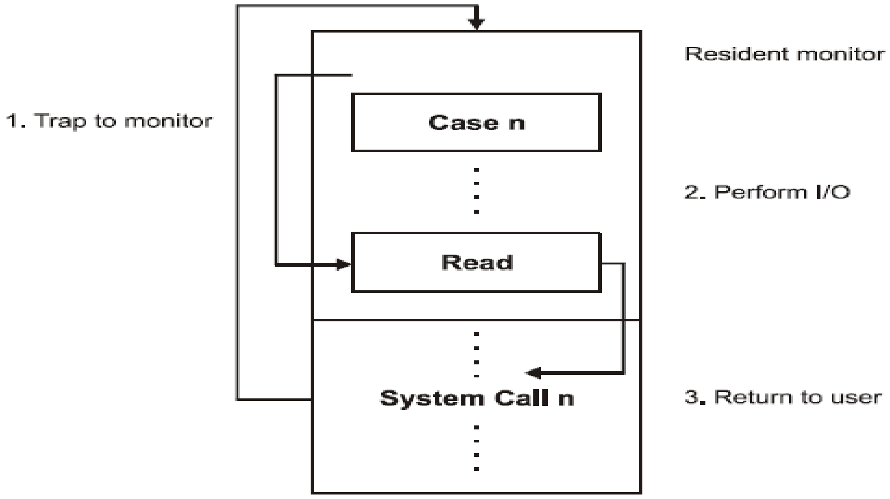


Fig.1.14: Example of I/O protection.

3. Memory protection

Usually process resides in address space which should have fix boundary so that the content of process can't interleave with other process contents. We must provide a protection to the process (at least for the interrupt vectored and the interrupt service routine). In order to provide protection to memory location the range is provided to the register to determine address space. In **figure 1.15** the job2 have boundaries with base and limit register value.

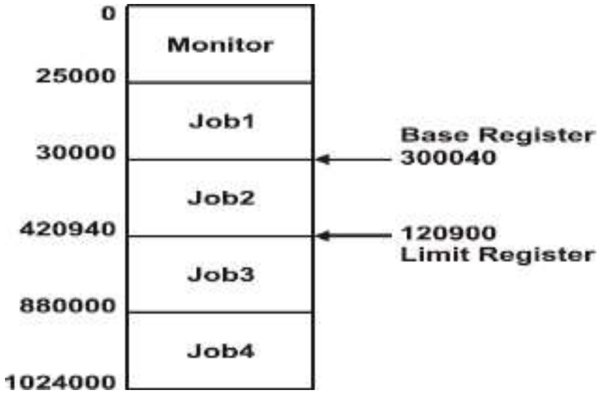


Fig.1.15: Memory protection example

Memory protection is a way to control memory access rights on a computer, and is a part of most modern instruction set architectures and operating systems. The main purpose of memory protection is to prevent a process from accessing memory that has not been allocated to it.

(i) Base Register: It hold the smallest legal physical address.

(ii) Limit Register: It consists of size of the range (max address).

Facts:-

- Memory outside the defined range is protected.
- No process can access the contents outside their address space.
- The address space should not less than base register and should not greater than limit register.

4. CPU protection

To prevent the user program getting stuck in an infinite loop and never returning control to OS uses the timer. After certain time, an interrupt occurs. Before turning control to user, OS loads up the timer to interrupt. Once the timer interrupt occurs, control goes back to OS. At this point OS may send out a fatal error message or may increase the timer limit. It is used in time sharing environments. In order to protect a, timer is provided.

- Timer is decremented at every clock tick.
- When timer reaches to zero then interrupt occurs.
- Fixed clock interval is required to allow the access of CPU to process.
- It also should ensure that CPU does not assigned infinite time to any process otherwise single process may get large time span.
- Load timer is a privileged instruction.

***Note:** All protection issues are uses to ensure that proper, fair and effective use of all resource to the process.*

1.12 Design and Implementation of OS

While design of any operating system many factors are needed to be consider, from both user and system prospective. The design and Implementation must be careful because if design fails so it may leads to produce bad/wrong O/P and will become reason to unsatisfied the user.

Basic Fundamental Principles:

1. Performance: For any product that must have perfect and should able to perform task with good manner and this is also big issues while OS designing.

2. Robustness: While any hazards in system, OS should be robust because if it collapses it cannot produce desired output and user could not get any satisfaction. Also system should be hard to break.

3. Efficiency: OS should be efficient to produce the O/P.

4. Scalability: OS should have scalability capability, if any user wants to increase the number of operations that time OS should support to user requirement up to some extents.

5. Predictability: It should not be unpredictable. Every time it should produce the same result.

6. Exclusive use of Resource: It's should provide fair and exclusive use of resource to each process. So that they can executes their task without interleaving to other process.

7. Modularity

8. Extensibility

9. Reliable

10. Simplicity and may more

(A) Design Goal

It is easier to define an operating system by what it does than what it is, but even this can be tricky. The primary goal of some of operating system is satisfied the user, efficient operation of the computer system. Operating systems and computer architecture have influenced each other to great deal. To facilitate the use of the hardware, researchers developed operating systems. Users of the operating system then proposed changes in hardware design to simplify the design. In this short historical review, it has been noticed that, how identification of operating-system problems led to the invention of new hardware features. While design of OS selects a best H/W type to system is one of the goals. The Nature of system (processing) like Batch processing, Time sharing system, etc. Design Goal of OS is categorized into two types.

(i) **User Goal:** As a user perspective OS should be:-

- Easy to used,
- Must be reliable,
- Safe, fast to produce any output,
- Easy to learn and may more as all user requirements.

(ii) System Goal: As a system perspective it also having same goal while design.

- It should be Error free.
- Easy to design, reliable, flexible.
- It should have fair maintenance.
- It should effective to produce results.

(B) Mechanism and policy

(i) Mechanism - Mechanism determine how to do something. Mechanism is simply, if we want to protect some resources then how that resource can be protected. E.g. In order to protect CPU we uses timer mechanism. Consider a mechanism for giving priority to certain type of programs over other. If mechanism is properly separated from policy, it can be used to support policy decision. Mechanism achieves the fair and protected use of CPU among the processes.

1. Policy – Policies are also one of the most important principles of OS design. The policy may vary place to place and time to time. Decision is important for all resource allocation reallocation and scheduling. E.g. how many times user will allow to uses a resource. In simple word policy is nothing but "***what to do***"?

(C) Implementation:

Once design and other criteria are set we need to think how to develop an OS which will be able to work properly. The implementation is a part of development of new OS by using the language paradigm. With the help of any language construct we can develop the OS. In earlier era the OS were developed in assembly level language but assembly level language has so many drawbacks. But now a day's C/C++ (oriented language) are the language that are commonly used. Still small blocks of assembly code are needed for development. Especially related to some low level I/O functions in device drivers, turning interrupt on and off and the Test and Set Instruction for Synchronization Facilities.

Higher level languages allow the code to be written faster. It also makes the OS much easier to port to different hardware platforms.

Drawback of Assembly Language

- Assembly language is difficult to code.
- It needs to keep extra information in consideration, like op-code etc.
- Assembly level language does not support to modularity.
- It also not has convenient to user, as it is hard to code.

High level language

- Now a day the high level language is used to build an OS (kernel).
- Especially C language is preferred to design of OS.
- High level language is easy to code, easy to understand.
- More compact to maintain.
- Easy to debug and easy to port.

(D) System structure

Design and implementation system structure is categorized into some parts. This design issues show how system looks like internal working. How different services can be developed to different level. There are some structures of OS for design purpose.

(1) Simple structure (3) Monolithic kernel

(2) Layered approach (4) Microkernel

(1) Simple structure

Traditional OS, like MS-DOS/old UNIX are have simple structure. The hardware protection dual mode is not present in simple structure. All block are treat as the simple i.e. either user or system mode. (Means there are no provision to switch the mode).

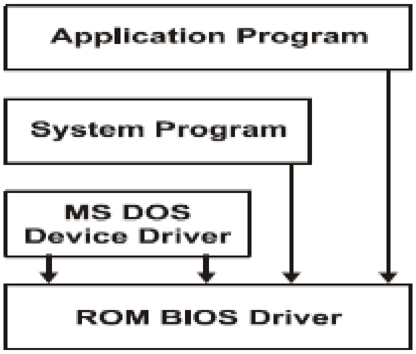


Fig.1.16: Simple Structure

The structure of OS is not separated and all services are provided in bulk of module. Even it is not possible to modify the structure to the user (constant structure, **figure 1.16**). Due to a single system structure all function are dependent on each others. If one fail so it affects others and they may lead to total crash. It is quite dependable on each other ultimately it take time to load and has low performance.

(2) Layered Approach

In order to overcome the drawback and provide separate functions at each level the layered approach was invented. In that the different layers are design to provide separate services. Very 1st layer (bottom) is an H/W layered which supports to H/W activity and their driver routines. Layered 0 is dedicated to process management Layered 1 is allocated to memory-management Layered 2 manages I/O device and buffering the information Layered n - etc.(**figure 1.17** shows idea), The fact is that, every layer provides services to each above it. At last upper layer is user interface layer that provides user interface to user for suitable use of OS, for every function.

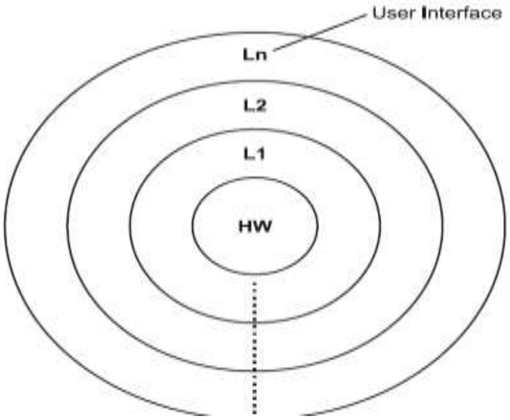


Fig.1.17: Layered Approach

Advantages

- Due to separation of layer and services it leads to modularity concepts.
- It allows developer to change inner working of any layer.
- It supports design of OS (because user get change to design develops some new thing).

Disadvantages:

- It is user oriented so the changes made by users, could not understand by others and also it's not required to others.
- Need careful definition of each layer at time of implementation.

Note: *[Micro Kernel and Monolithic Kernel structure have been discussed in kernel topic so it needs not to repeat here, author are requested to refer that section, i.e., kernel structure* ***figure 1.10 and 1.11****]*

Buffering and Spooling

Computer processes data, stores it, and sends it to different peripherals and output devices. Computer science uses a few terms specific to this type of process: spooling, buffering. These terms have specific meaning in computer science, but they're all having similar functions. Many use them interchangeably. The specifics depend on the case.

Buffering is a method of overlapping the computation of a job with its execution. It temporarily stores input or output data in an attempt to better match the speeds of two devices such as fast CPU and slow disk drive. If, for example, the CPU writes information to the buffer, it can continue its computation while the disk drive stores the information.

Buffering only allows the I/O of a job to overlap with its own computation. Watching video on YouTube by clicking "Play" will cause the video to start playing immediately. These subsequent parts are queued up in a buffer so that the video plays smoothly even though it's not fully downloaded when you start it.

With spooling, the disk is used as a very large buffer. Usually complete jobs are queued on disk to be completed later. A typical example is the spooler for a printer. When a print job is issued, the spooler takes care of it, sending it to the printer if it is not busy, or storing it on disk otherwise. Spooling allows the I/O of one job to overlap the computation of another. Spooling's name comes from the acronym for Simultaneous Peripheral Operation On-Line (SPOOL). Spooling waits until the entire operation is done before sending it to the output device or a network, and your likeliest encounter with spooling probably comes from sending a document to a printer.

Spooling helps to solve the problem of speed mismatch among different devices. For example CPU operates at a very high speed than the printer.

Note: - *Spooling is better than Buffering all the way. In Spooling, CPU Allows overlapping of one Job with the computation and output of other job...whereas in Buffering, the CPU overlaps input, output and processing of a single Job. (Job = Program)*

1.13 Case Studies:

1.13.1 Android

Operating Systems have developed a lot in last 15 years. Starting from black and white phones to recent smart phones or mini computers, mobile OS has come far away. Especially for smart phones, Mobile OS has greatly evolved from Palm OS in 1996 to Windows pocket PC in 2000 then to Blackberry OS and Android. One of the most widely used mobile OS these days is ANDROID. Android is a software bunch, comprising not only operating system but also middleware and key applications. The Android OS is one of the most commonly used mobile OS and there are plenty of devices on the market powered by this operating system.

Android OS is a Linux-based open source platform for mobile cellular handsets developed by Google and the Open Handset Alliance. The Android platform includes an operating system based upon Linux, GUI, Web browser and end user applications that can be downloaded. Although initial demonstrations of Android featured a generic QWERTY Smartphone and large VGA screen, the operating system was written to run on relatively inexpensive handsets with conventional numerical keypads. **Figure 1.18** shows popular icon of android operating system.

Figure 1.18 Android icon

Features & Specifications

- Android is a powerful Operating System supporting a large number of applications in Smart Phones.
- These applications make life more comfortable and advanced for the users.
- Hardware that supports Android is mainly based on ARM architecture platform.

Applications

Following are the basics application of Android operating system:

- Android applications are composed of one or more application components (activities, services, content providers, and broadcast receivers)
- Each component performs a different role in the overall application behavior, and each one can be activated individually (even by other applications)
- The manifest file must declare all components in the application and should also declare all application requirements, such as the minimum version of Android required and any hardware configurations required
- Non-code application resources (images, strings, layout files, etc.) should include alternatives for different device configurations (such as different strings for different languages)

1.13.2 UNIX:

Is a popular multiuser, multitasking, portable operating system developed at Bell Lab in the early 1970. UNIX was one of the first operating system to be written in high level programming language. This meant that it could be installed on any computer wherever C compiler exists. UNIX is an open source language it means any user can get the hardcode of operating system and can modify it, as per their requirement, and can built new featured operating version. UNIX systems also have a graphical user interface (GUI) similar to Microsoft Windows which provides an easy to use environment. However, knowledge of UNIX is required for operations which aren't covered by a graphical program, or for when there is no windows interface available, for example, in a telnet, ssh (Secure Shell) session.

Files and processes

Everything in UNIX is either a file or a process. A process is an executing program identified by a unique PID (process identifier). A file is a collection of data. They are created by users using text editors, running compilers etc.

Examples of files:

- A document (report, essay etc.)
- The text of a program written in some high-level programming language
- Instructions comprehensible directly to the machine and incomprehensible to a casual user, for example, a collection of binary digits (an executable or binary file);
- A directory, containing information about its contents, which may be a mixture of other directories (subdirectories) and ordinary files.

The Directory Structure:-

All the files are grouped together in the directory structure. The file system is arranged in a hierarchical structure, like an inverted tree. The top of the hierarchy is traditionally called root (written as a slash /)

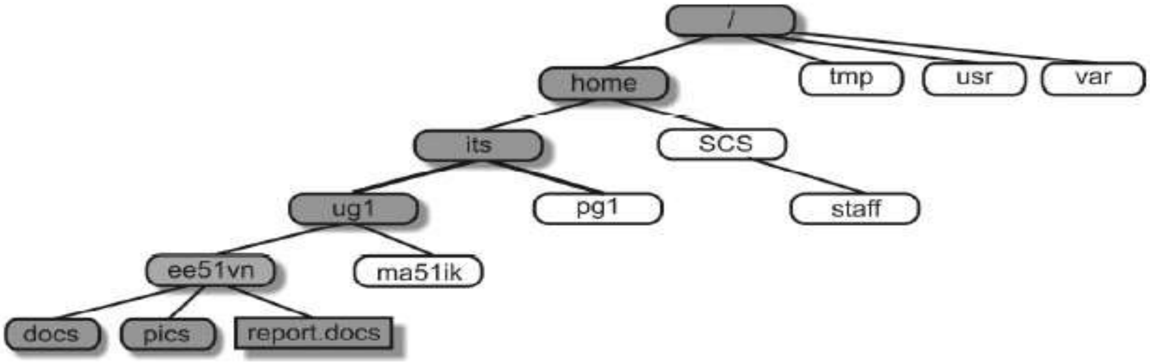


Fig.1.19: Directory structure

In the **figure 1.19**, we can see that the home directory of the undergraduate student "ee51vn" contains two sub-directories (docs and pics) and a file called report.doc. The absolute path to the file report.doc is **"/home/its/ug1/ee51vn/report.doc"**

UNIX OS comprise of:-

Kernel: - schedule task, Manage data /file and storage, Enforce security mechanism, perform all hardware access

Shell: - presents each user with a prompt, Interrupt command types by a user, Executes user commands.

Utility: - File management, user management, process management, and printing.

Features:

1. **Multiuser**:-more than one user can use machine at a time support via terminals.
2. **Multitasking**: - more than on program can be run at a time.
3. **Hierarchy directory structure**:-to support the organization and maintenance of file.
4. **Portability**:-Only the kernel written in assembler tools for program development a wide range of support tools (debuggers, compilers)

Distribution of UNIX:

There are many different version of UNIX, although they share common similarities, the most popular varieties of UNIX are Sun Solaris, GNU / Linux and MacOS X. Some distributions are fedora, red- hat, open-suse, mandriva, cent-OS and many more.

Note: *It is very difficult to accommodate/explain/ define UNIX in few lines because this operating system is has huge scope; students are suggested to read more on UNIX forum.*

1.13.3 Windows 8

Microsoft provided support for Windows XP for the past 12 years. But the time came for us, along with our hardware and software partners, to invest our resources toward supporting more recent technologies so that we can continue to deliver great new experiences. As a result, technical assistance for Windows XP is no longer available, including automatic updates that help protect your PC. Microsoft has also stopped providing Microsoft Security Essentials for download on Windows XP platform. If you already have Microsoft Security Essentials installed, you'll continue to receive antimalware signature updates for a limited time.

However, please note that Microsoft Security Essentials (or any other antivirus software) will have limited effectiveness on PCs that do not have the latest security updates. This means that PCs running Windows XP will not be secure and will still be at risk for infection. Windows 8 is a completely redesigned operating system that has been developed from the ground up with touch screen use in mind as well as near-instant-on capabilities that enable a Windows 8 PC to load and start up in a matter of seconds rather than in minutes. Windows 8 will replace the more traditional Microsoft Windows OS look and feel with a new "Metro" design system interface that first debuted in the Windows Phone 7 mobile operating system.

The Metro user interface primarily consists of a "Start screen" made up of "Live Tiles," which are links to applications and features that are dynamic and updated in real time. Windows 8 supports both x86 PCs and ARM processors. Windows 8 represents a major departure from previous versions of Windows because it is based on the Metro design language, which facilitates a touch screen UI (user interface) similar to the ones found in mobile phones and tablet computers. Microsoft's Surface tablet, released in early 2013, runs Windows 8.

The basic version of Windows 8, suitable for most users, works with touch screen devices as well as with traditional keyboard-and-mouse systems. A version called Windows RT supports the ARM processor architecture. Another version, Windows 8 Pro, is aimed at business and technical professionals who desire encryption, virtualization, PC management and domain connectivity. The Windows 8 screen is tile-based, with each tile representing a specific application. The taskbar has no Start button, although users can bring up a Start screen by touching or clicking the lower-left corner of the display. The Windows 8 Release Preview contains a Metro version of Internet Explorer 10, along with application for news, sports, and travel. It also offers an updated Windows Explorer, an updated Task Manager, and picture password capability.

Major feature:-

- Upgrades from Windows 7
- Windows Store
- Connected standby
- Microsoft account

- Language switching "on the fly"
- Windows Update
- Windows Defender (anti-malware program)
- Support for multiple displays
- Mobile broadband capability
- Reset and refresh

Note: *All three case studies have separate existence so it is very difficult to illustrate in one page students are requested to refer any web solution.*

A **Virtual Machine** (VM) is a compute resource that uses software instead of a physical computer to run programs and deploy apps. One or more virtual “guest” machines run on a physical “host” machine. Each virtual machine runs its own operating system and functions separately from the other VMs, even when they are all running on the same host. This means that, for example, a virtual MacOS virtual machine can run on a physical PC.

- A VM (virtual machine) is an emulation of a computer system, where these machines use computer architectures to provide the functionality of a physical computer.
- The physical device on which virtual machines work is known as Host, whereas the virtual machines are known as Guest.
- A single host can have multiple numbers of guests.

Virtual machine technology is used for many use cases across on-premises and cloud environments. More recently, public cloud services are using virtual machines to provide virtual application resources to multiple users at once, for even more cost efficient and flexible compute.

What are virtual machines used for?

Virtual machines (VMs) allow a business to run an operating system that behaves like a completely separate computer in an app window on a desktop. VMs may be deployed to accommodate different levels of processing power needs, to run software that requires a different operating system, or to test applications in a safe, sandboxed environment.

Virtual machines have historically been used for server virtualization, which enables IT teams to consolidate their computing resources and improve efficiency. Additionally, virtual machines can perform specific tasks considered too risky to carry out in a host environment, such as accessing virus-infected data or testing operating systems. Since the virtual machine is separated from the rest of the system, the software inside the virtual machine cannot tamper with the host computer.

How do virtual machines work?

The virtual machine runs as a process in an application window, similar to any other application, on the operating system of the physical machine. Key files that make up a virtual machine include a log file, NVRAM setting file, virtual disk file and configuration file.

Advantages of virtual machines

Virtual machines are easy to manage and maintain, and they offer several advantages over physical machines:

- VMs can run multiple operating system environments on a single physical computer, saving physical space, time and management costs.
- Virtual machines support legacy applications, reducing the cost of migrating to a new operating system. For example, a Linux virtual machine running a distribution of Linux as the guest operating system can exist on a host server that is running a non-Linux operating system, such as Windows.
- VMs can also provide integrated disaster recovery and application provisioning options.

Disadvantages of virtual machines

While virtual machines have several advantages over physical machines, there are also some potential disadvantages:

- Running multiple virtual machines on one physical machine can result in unstable performance if infrastructure requirements are not met.
- Virtual machines are less efficient and run slower than a full physical computer. Most enterprises use a combination of physical and virtual infrastructure to balance the corresponding advantages and disadvantages.

The two types of virtual machines

Users can choose from two different types of virtual machines—process VMs and system VMs:

A process virtual machine allows a single process to run as an application on a host machine, providing a platform-independent programming environment by masking the information of the underlying hardware or operating system. An example of a process VM is the Java Virtual Machine, which enables any operating system to run Java applications as if they were native to that system.

A system virtual machine is fully virtualized to substitute for a physical machine. A system platform supports the sharing of a host computer's physical resources between multiple virtual machines, each running its own copy of the operating system. This virtualization process relies on a hypervisor, which can run on bare hardware, such as VMware ESXi, or on top of an operating system.

What are 5 types of virtualization?

All the components of a traditional data center or IT infrastructure can be virtualized today, with various specific types of virtualization:

- **Hardware virtualization**: When virtualizing hardware, virtual versions of computers and operating systems (VMs) are created and consolidated into a single, primary, physical server. A hypervisor communicates directly with a physical server's disk space and CPU to manage the VMs. Hardware virtualization, which is also known as server virtualization, allows hardware resources to be utilized more efficiently and for one machine to simultaneously run different operating systems.
- **Software virtualization**: Software virtualization creates a computer system complete with hardware that allows one or more guest operating systems to run on a physical host machine. For example, Android OS can run on a host machine that is natively using a Microsoft Windows OS, utilizing the same hardware as the host machine does. Additionally, applications can be virtualized and delivered from a server to an end user's device, such as a laptop or smartphone. This allows employees to access centrally hosted applications when working remotely.
- **Storage virtualization**: Storage can be virtualized by consolidating multiple physical storage devices to appear as a single storage device. Benefits include increased performance and speed, load balancing and reduced costs. Storage virtualization also helps with disaster recovery planning, as virtual storage data can be duplicated and quickly transferred to another location, reducing downtime.
- **Network virtualization**: Multiple sub-networks can be created on the same physical network by combining equipment into a single, software-based virtual network resource. Network virtualization also divides available bandwidth into multiple, independent channels, each of which can be assigned to servers and devices in real time. Advantages include increased reliability, network speed, security and better monitoring of data usage. Network virtualization can be a good choice for companies with a high volume of users who need access at all times.
- **Desktop virtualization**: This common type of virtualization separates the desktop environment from the physical device and stores a desktop on a remote server, allowing users to access their desktops from anywhere on any device. In addition to easy accessibility, benefits of virtual desktops include better data security, cost savings on software licenses and updates, and ease of management.

Important Questions:-

Que.: Which of the following instruction should be privileged?

The meaning of privileged is that having rights to do something to the different activity.

(a) Set mode to kernel mode.

Ans.: Yes

(b) Reboot

Ans.: Yes

(c) Read the program states word

Ans.: No

(d) Disable interrupt

Ans.: Yes

(e) Write the instruction register

Ans.: No

(f) Set the value of timer

Ans.: Yes

(g) Read the clock

Ans.: yes

(h) Clear memory

Ans.: yes

(i) Turn off interrupts

Ans.: yes

(j) Switch from user to monitor mode.

Ans.: No.

Que.: What are the different categories of system S/W.

There are so many categories of system S/W as per their uses to specific task.

1. Operating system: OS itself is one of the system S/W, which has ability to control all computer system with proper control.

2. Boot loader: It is a piece of code that runs before any OS load in memory. Loader helps to load system S/W and application S/W. eq. Grub, LILO, etc.

3. Compiler: - Compiler is system S/W that help to transfer/conversion of code in specific language. Basically it converts the High level code into low level (M/C).

4. Device driver:- Each device having its own device driver which helps to run respective device. Driver is H/W dependent and OS specific. When calling program invokes a routine in the driver, the driver issues commands to device. The drivers typically communicated with device through system bus.

5. Assembler, Compiler, Interpreter: are the translators

Note: *There are also many system software are available to specific use that beyond the syllabi.*

Que.: Justify Your Answer:

1. Operating system acts as government.

Ans.: Yes

Justification:

As OS control all activities of resources and processes, which also provides all services to user as well as to the system. Especially OS provide resource management like CPU, Memory I/O etc. As like government has overall control on each component/person similarly OS also has control to computer system. Finally OS is ***"all in all".***

2. Hard disk is an Output device or Input device.

Ans.: Hard disk is both I/P and O/P device.

Justification: Hard disk contains data, file, and information.

Case 1:- when user want to read data which is stored on disk that time Hard disk called input device.

Case 2:- when user wants to store data on disk that time Hard disk is called as a O/P device.

Note: *Some time hard disk called neither input nor output device.*

3. Time sharing does not support multiprogramming.

Ans.: Yes/True

Justification: Time sharing means at particular time t1 only one process can be executed by CPU, on the basis of time quantum. In time sharing the CPU allocated to process only if program is in memory. Whereas multiprogramming means more than one program that can available in memory. And in multiprogramming the decision of task assigning is done by OS itself.

4. Multiprocessing implies multitasking.

Ans.: Justification: In multiprocessing environment more than one physical processor are available to perform task. But in multiprocessing environment multiple task can done at a time on some or different processor (No issues). While multitask is a concept which uses the context switching to switch from one task to another task.

By above points, with availability of CPU it’s possible to perform multiple tasks on the single processor.