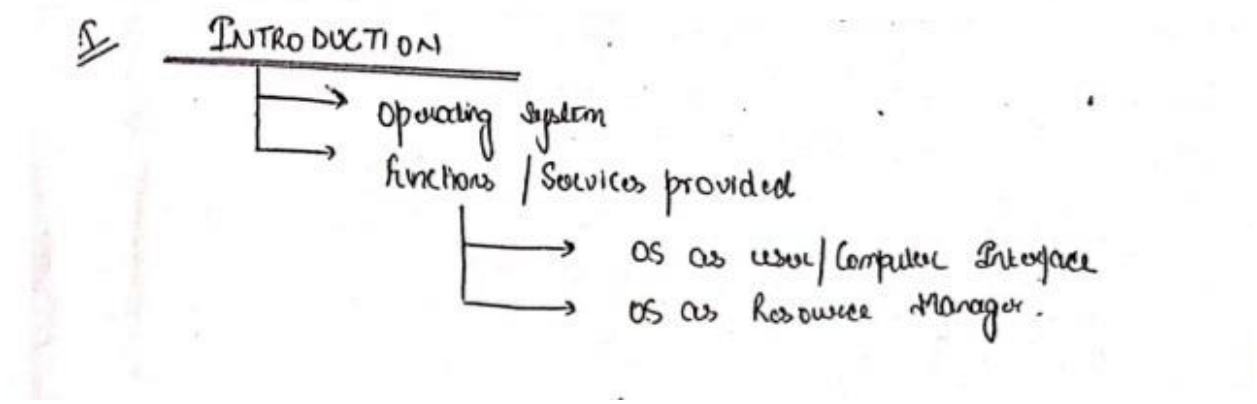
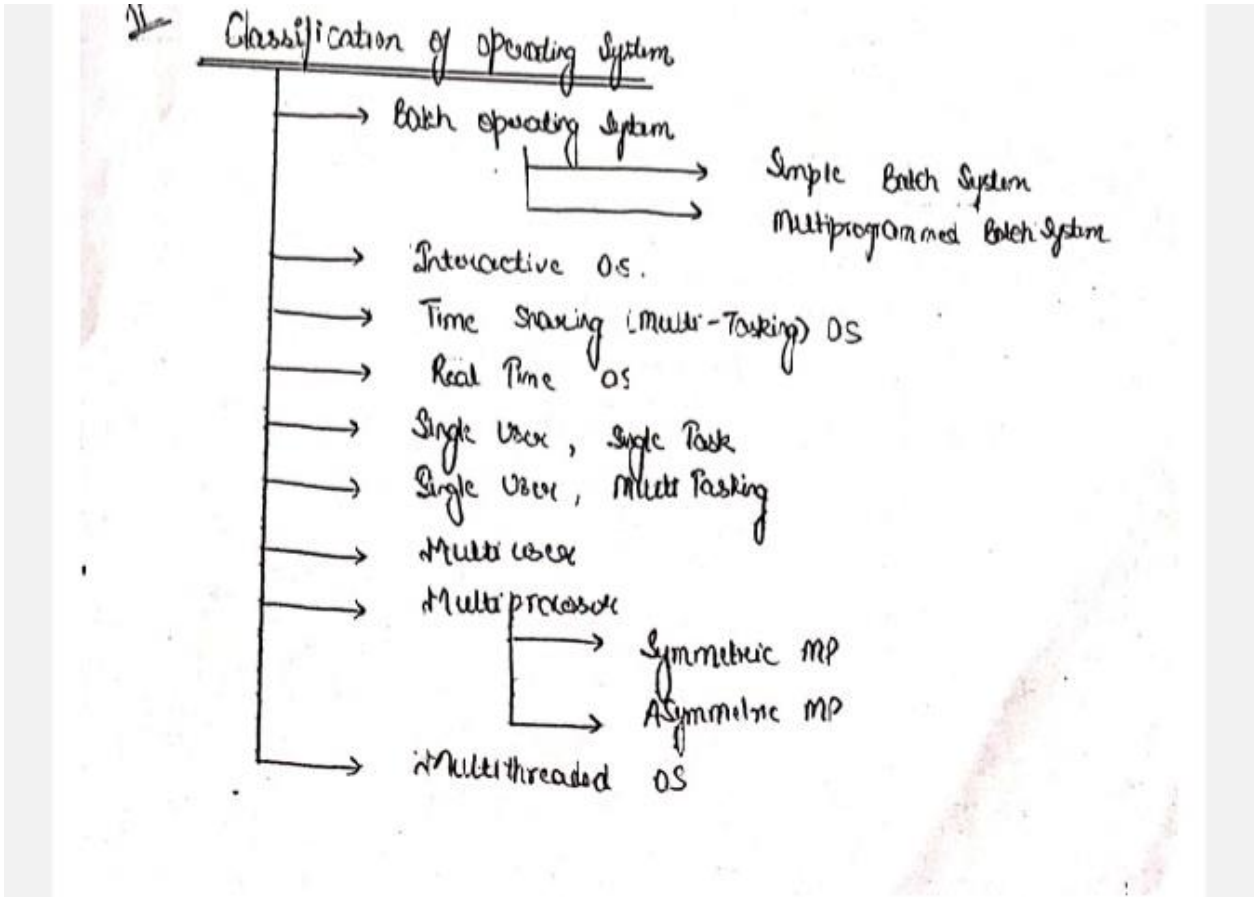
UNIT – 1
BCS-401: Operating System

UNIT – 1

Introduction: Operating system and functions, Classification of Operating systems- Batch, Interactive, Time sharing, Real Time System, Multiprocessor Systems, Multiuser Systems, Multiprocessor Systems, Multithreaded Systems, Operating System Structure- Layered structure, System Components, Operating System services, Reentrant Kernels, Monolithic and Microkernel Systems.





III. Operating System Structure

- Layered Structure
- Monolithic OS
- Micro - Kernel or Client - Server

IV. Operating System Components

- Process Mgmt
- Main Memory Mgmt
- File Mgmt.
- Secondary Storage Mgmt
- User - Interface
- I/O System Mgmt
- Networking
- Protection System
- Command Interpreter System

V. Operating System Services

- Program Execution
- I/O operation
- File System Manipulation
- Communications
- Error Detection
- Resource Allocation
- Accounting
- Protection.

VI. RE-ENTRANT KERNELS

VII. EXTRA

- Important Points
- Last year U.P.T.U Question.
- Assignment
- Tutorial.

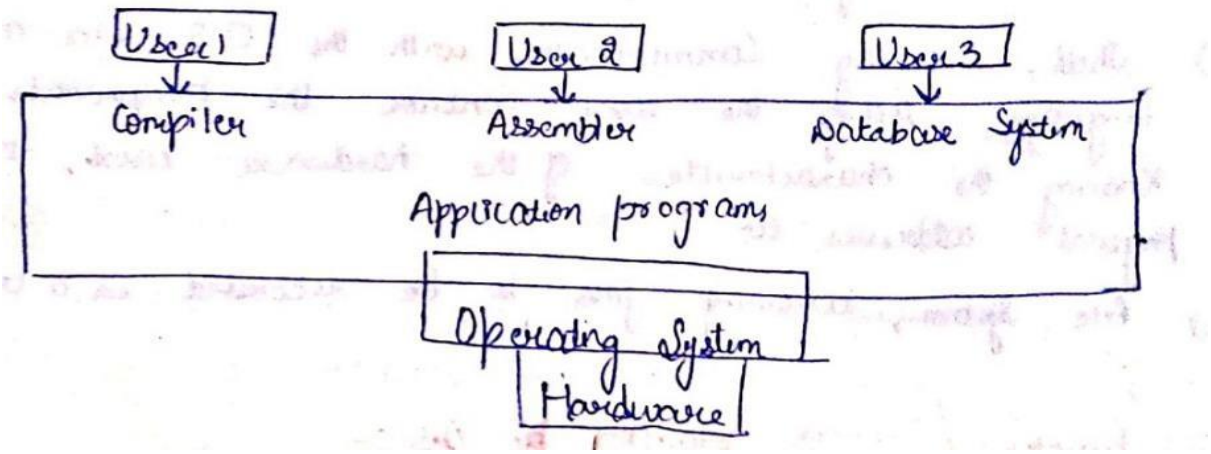
I. INTRO: Operating System & Components

(i) Operating System

An Operating System is a program that controls the execution of application programs and acts as an interface between a user of a computer and computer hardware.

Operating System are the essential part of any computer system. It acts as an intermediate between user of a computer and computer hardware.

It controls and coordinates the use of hardware among the various application programs for the various users.



Abstract View of Components of Computer System

Purpose: To provide an environment in which a user can execute programs.

Goal: To make the computer system Convenient to use.

To use the Computer hardware in an efficient manner.

Components of Operating System:

The operating system comprises a set of the software packages that can be used to manage interactions with the hardware. The following elements are generally included in this set of software:-

(a) **Kernel:** The Kernel is a program that constitutes the central Core of computer operating system. A kernel can be contrasted with a shell, which is the outer most part of an operating system and a program that interacts with user commands.
The Kernel itself does not interact directly with the user, but rather interacts with the shell and other programs.
The Kernel is the first part of the operating system to load into memory during booting and it remains there for the entire duration of the computer session because its services are required continuously.

It typically includes-

- **A Scheduler,** which determine how the various processes shares the kernels processing time.
- **An interrupt Handler,** which models all request from the various hardware devices that compete for the Kernels services.
- **A Supervisor ,** which grants use of the computer to each process when it is scheduled.

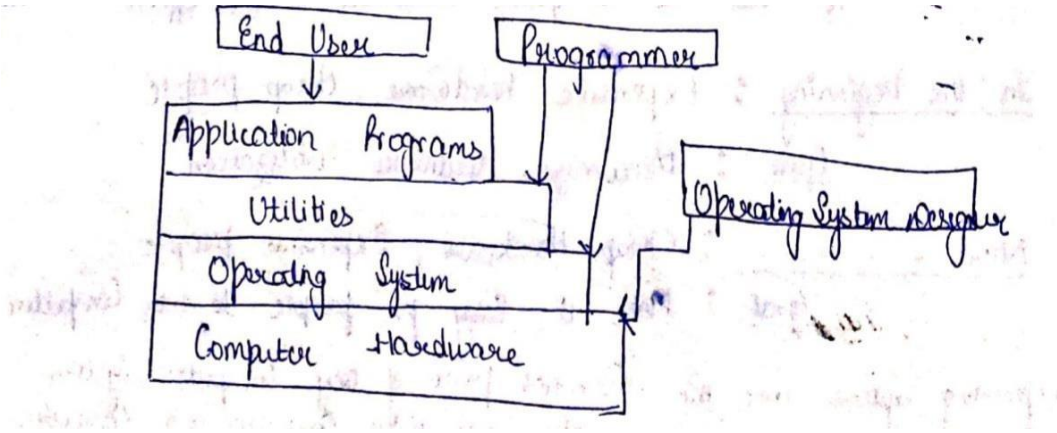
(b) **Shell**, Allowing communication with the O.S via a control language, letting the user control the peripherals without knowing the characteristics of the hardware used, management of physical addresses etc.

(c) **File system**, allowing files to be recorded in a tree structure.

(ii) Functions / Services provided by Operating System

(a) O.S as a user / Computer Interface:

The Hardware and software used in providing applications to a user can be viewed in a layered or a hierarchical fashion.



Layers & Views of Computer System

User of those applications, the end user, generally is not concerned with the details of computer hardware. Thus, the end user views the computer system in terms of set of applications. Basically, OS typically provides services in the following areas:

ProgramDevelopment–OSprovidesavarietyoffacilitiesandservicessuchas editors and debuggers, to assist the programmers in creating programs.

Access to I/O devices: Each I/O devices requires its own peculiar set of instructions or control signals for operation.

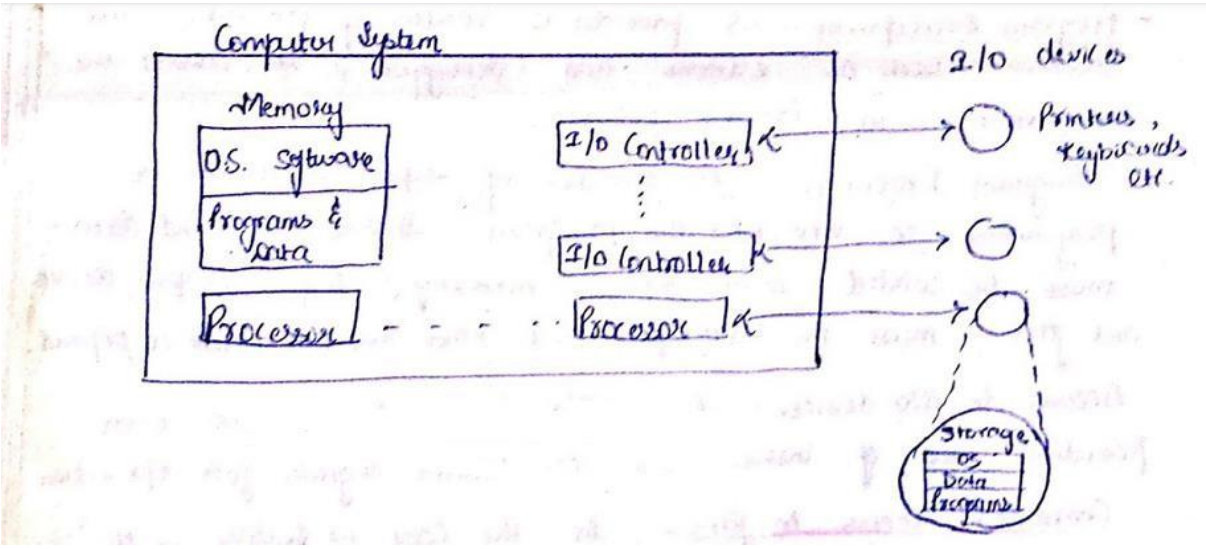
Control access to files-In the case of system with multiple users, the OS may provide protection mechanism to control access to the files.

Error detection & Response – A variety of errors can occur while a computer system is running. These include internal and external hardware error (such as memory error or device failure) and software errors such as division by zero.

Accounting –A good OS will collect usage statistics for various resources and monitor performance parameters such as response time.

(b) The OS as a Resource Manager –

A Computer is a set of resources for the movement, storage and processing of data and for the control of these functions. The OS is responsive for managing these resources. A portion of OS is in main memory. This includes the Kernel, which contains the most frequently used functions in the OS and at a given time, other portions of the OS currently in use. The remainder of the main memory contains user programs.



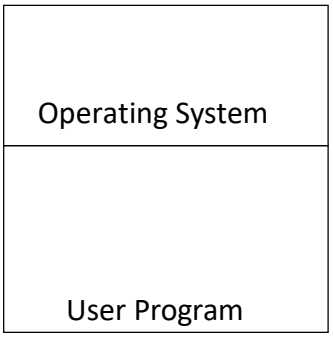
OS as a Resource Manager

II. Classification of Operating System

(a) Batch Operating System

(a.1) Simple Batch System-

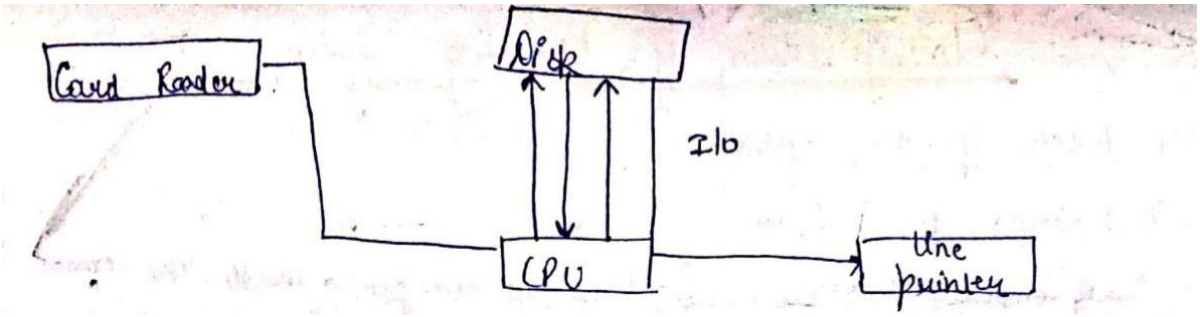
Early computer were enormously large m/c run from a console. The common input Devices were card reader and tape drives. The common output device were line printers, tape drives and card punches. The user of such system did not interact directly with the computer system. Rather, the user prepared a job which consisted of the program, the data, and some control information about the nature of the job-andsubmitteditto thecomputeroperator.Thejobwouldusuallybeintheformof punch cards. At some later time, the output appeared. The output consisted of the result of the program, as well as dump of memory and registers in the case of program error.
To speed up processing, jobs with similar needs were batched together and were run through the computer as a group. Thus, the programmers would leave their programs with the operator. The operator would sort programs into batches with similar requirements and as the computer become available, would run each batch. The output from each job would we sent back to the appropriate programmer.



Memory Layout for a simple batch system.

The definite feature of the batch system is the lack of interaction between the user and the job while that job is executing. The job is prepared and submitted and at some later time the output appears.
The delay between job submission and job completion may result from the amount of computing needed or from delays before the operating system starts to process the job.
In the execution environment, the CPU is often idle. This idleness occurs because the speeds of the mechanical I/O devices are intrinsically slower than those of electronic devices. Even a slow CPU works in the microsecond range, with thousands of instructions executed per second. A fast card reader, on the other hand, might read 1200 cards per minute. Thus, the difference is speed between the CPU and I/O devices maybe three orders of magnitude or more.
The introduction of disk technology has help in this regard. Rather than the cards being read from the card reader directly into memory, and then the job is being processed, card are read directly from the card reader onto the disk. The location of the card images in recorded in a table kept by the operating system. When a job is executed, the operating system satisfies its request for card reader input by reading from the disk. Similarly, when the job request the printer to output a line that line is copied into

a system buffer and is written to the disk. When the job is completed, the output is actually printed. This form of processing is called is spooling. The name is an acronym for simultaneous peripheral operation online, spooling is essence uses the disk as a huge buffer, for reading as far ahead as possible on input devices and for sorting output files until the input devices are able to accept them.



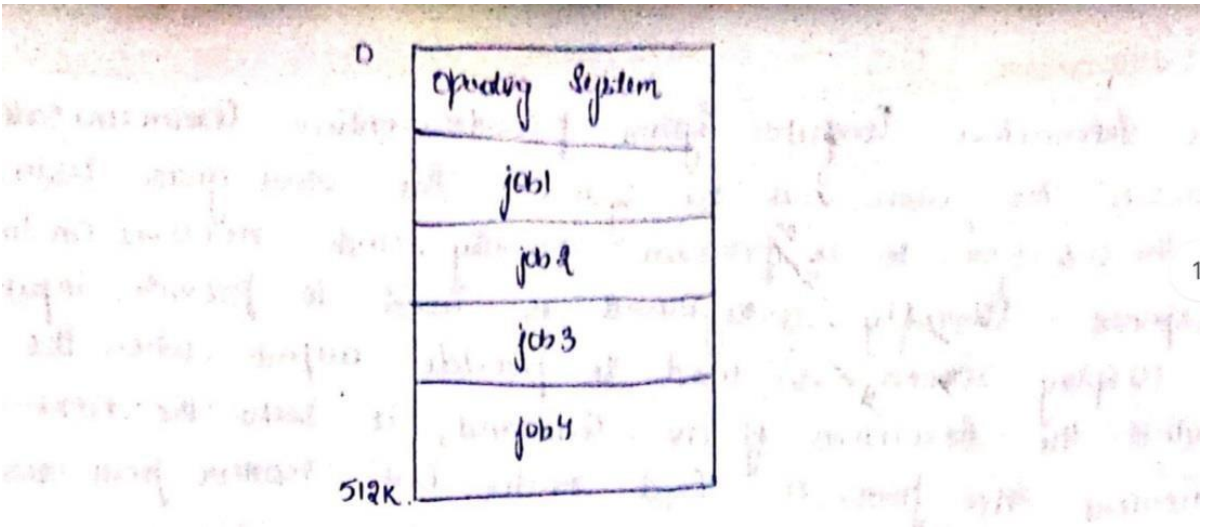
Spooling

Spooling is also used for processing data at remote sites. The CPU sends the data via communication path to a remote printer. Thus more processing is done at its own speed, with no CPU intervention. The CPU just needs to be notified when the processing is completed, so that it can spool the next batch of data.

(a.2) Multi programmed Batch System-

Spooling provides and important structure called job pool. Spooling will generally result in several jobs that have already being waiting on disk ready to run. Jobs must be run sequentially on a First Come First serve basis. However when several jobs are on direct access device, such as a disk job scheduling becomes possible.
The most important aspect of job schedule is the ability of multiprogramming.

Idea: The OS keeps several jobs in memory at a time. The OS picks and begins to execute one of the jobs in memory. Eventually, the job may have to wait for some task, such as tape to be mounted on an I/O operation to complete.
In a non – multi programmed system the CPU would sit idle.
In a multiprogramming system, the OS simply switches is to and execute another job.



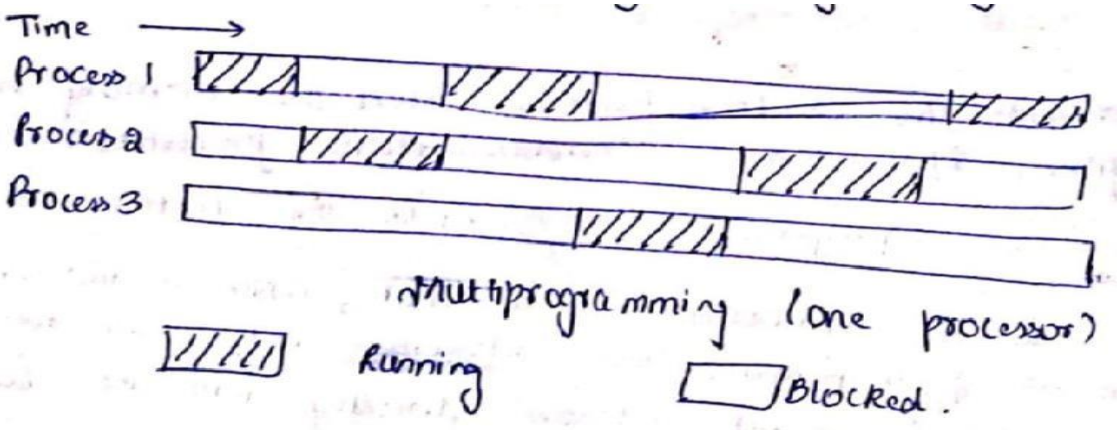
When the job needs to wait, the CPU switched to another job and so on. Eventually the first job finishes waiting and gets the CPU back. As long as there is always some job to execute, the CPU will never be idle.

Multi programming is a first instance when the OS must make decisions for the users. All the jobs that enter the system are kept in the job pool. This pool consist of all processes residing on mass storage awaiting allocation of main memory. If several jobs are ready to be brought in memory and there is not enough room for all of them, then the system must choose among them, making the decision is job scheduling.

When the OS selects a job from the job pool, it loads the job into memory for execution. Having several programs in memory at the same time requires having some form of memory management.
Finally, multiple jobs running concurrently require that their ability to affect one another be limited in all phase of the OS.

Multi Programming:-

A mode of operation that provides for the interleaved execution of two or more computer program by a single processor.



(b) Interactive OS-

Interactive computer system provide online communication between the user and the system. The user give instructions to the OS or to a program directly, and receives an immediate response. Usually, a keyboard is used to provide input and the display screen is used to provide output. When the OS finish is execution of one command, it seeks the next control statement not from our card reader, but rather from users' keyboard. The user gives a command, waits for the response and decides on the next command based on the result of previous one.

(c) Time Sharing (Multi-Tasking) System–

Time Sharing or Multi-tasking is a logical extension of multiprogramming. Multiples job are executed by the CPU switching between them, but the switches occurs so frequently that the users may interact with each program while it is running.
They were developed to provide interactive use of a computer system at reasonable cost. A time shared OS uses CPU scheduling and multi programming to provide each user with a small portion of a time shared computer. Each user has at least one

separate program in memory.
A time shared OS allows many users to share the computer simultaneously. Since each action or command in a time shared system tends to be short, only a little CPU time is needed for each other.
As the system switches rapidly from one user to the next so rapidly, each user is given the impression that she has her own computer, whereas actually one computer is being shared among uses.

(d) Real Time System-

Real time system are used when there are rigid time requirement on the operation of a processor.
It has well defined, fixed time constraints. Processing must be done with in the defined constraints or the system will fail.
It is considered to function correctly only if it returns the correct result within any time constraints.
Contrast this requirement to a time sharing system where it is desirable to respond quickly, or to a batch system, where there may be no time constraints at all. System that control scientific experiments, medical imaging system and some automobile engine fuel injection system etc.

There are two types of Real Time systems-

(e.1) Hard real time systems -It guarantees that critical task complete on time.

(e.2) Soft Real Time System–Where a critical task gets priority over other task and retains that priority until it completes.

(e)Single user, Single task-

This OS is designed to manage the computer so that one user can effectively do one thing at a time. The Palm OS for Palm handled computer is an example of this.

(f) Single User Multi-tasking (multi-purpose)-

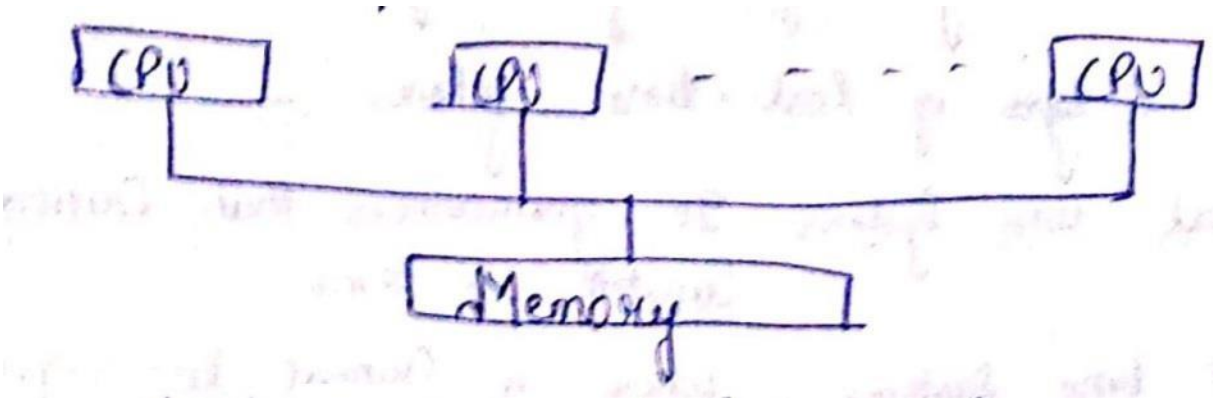
This is a type of OS most people use on their desktop and laptop today. Example- Microsoft Windows and Apple Mac OS, they will let us single user have several programs in operation at the same time.

(g) Multi-User-

Multi- User OS allows many different users to take advances of the computer resource simultaneously. The OS must make sure that the requirement of the various users are balanced, and that each of the program they are using has sufficient and separate resources so that a problem with one user does not affect the entire community of users. Example- Unix, UMS.

(h) Multi-processor System-

A multiprocessor OS is one that can run computer system that contain more than 1 processor.



Advantage-Increased throughput, Increased Reliability, Economical.

They are of two types-

(h.1) Symmetric Multiprocessing (SMP)-

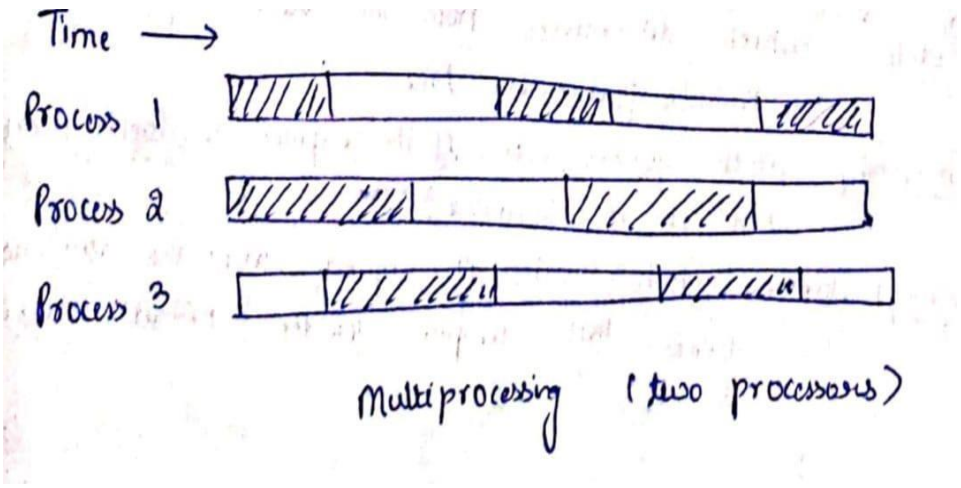
Each processor seems on identical copy of OS.
Most modern OS support SMP.
Many processors can run at once without performing deterioration.
Tightly coupled system-processor share memory and a clock.

(h.2) Asymmetric Multiprocessing (ASMP)-

Each processor is assigned specific specific task, Master processor schedules and allowed work to slave processor.
More common in extremely large systems.

Multi processing:-

A mode of operation that provide for parallel processing by two or more processor of a multi-processor.



.

(f) Multithreaded OS-

An OS is known as multithreaded when several “task” may be run at the same time. The application consists of a sequence of instructions known as “threads”. These threads will be at alternatively active, on standby suspended or destroyed, according to the priority assigned to them and may be run simultaneously.
Example- Linux, UNIX, Windows 2000

On a single processor, multi-threading generally occurs by time division multiplexing, the processor switches between different threads. The context switching generally happens frequently enough that the user perceives the threads or task as running at the same time.
On a multiprocessor multi-care system, the threads or tasks will actually run at the same time, with each processor or core running a particular thread or task.

Advantages–

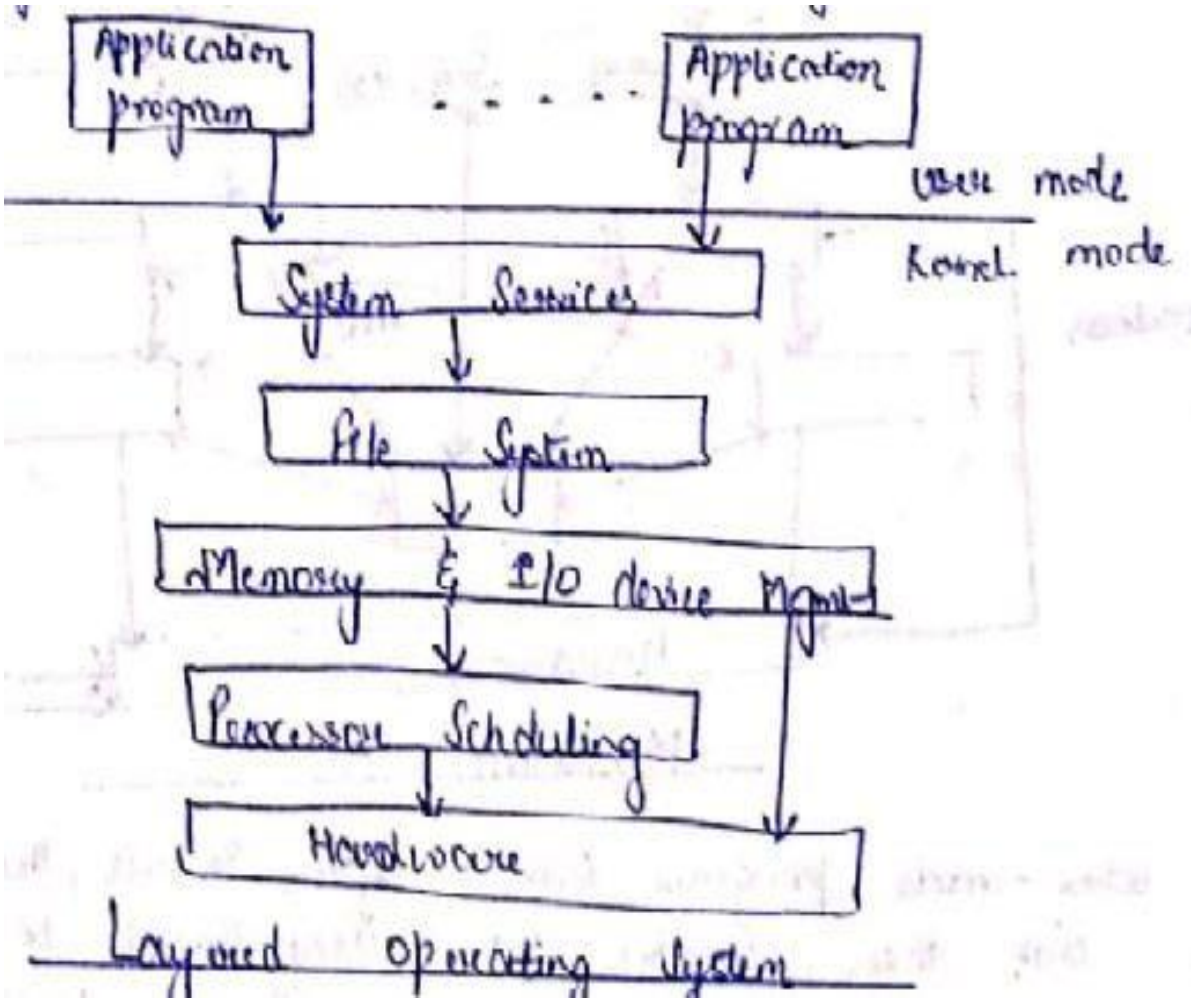
Responsive.
Resource Sharing.
Economy.
Utilization of multiprocessor architectures.

III. OPERATING SYSTEM STRUCTURE

On the basis of their structuring mechanism, operating system are broadly classified into 3 categories-

(a) Layered Structure–

The component of layered OS are organized into modules and buyers them one on top of the other. Each module provide a set of function that other module can call. Interface functions at any particular level can invoke services provided by inner layers but not the other way around.



Advantage: Each layer of code is given access to only the lower level interfaces it requires, thus writing the amount of code wields unlimited power i.e. in thus approach, the Nth layer can access services provided by the (N-1)th layer and provided services to the (N+1)th byer.

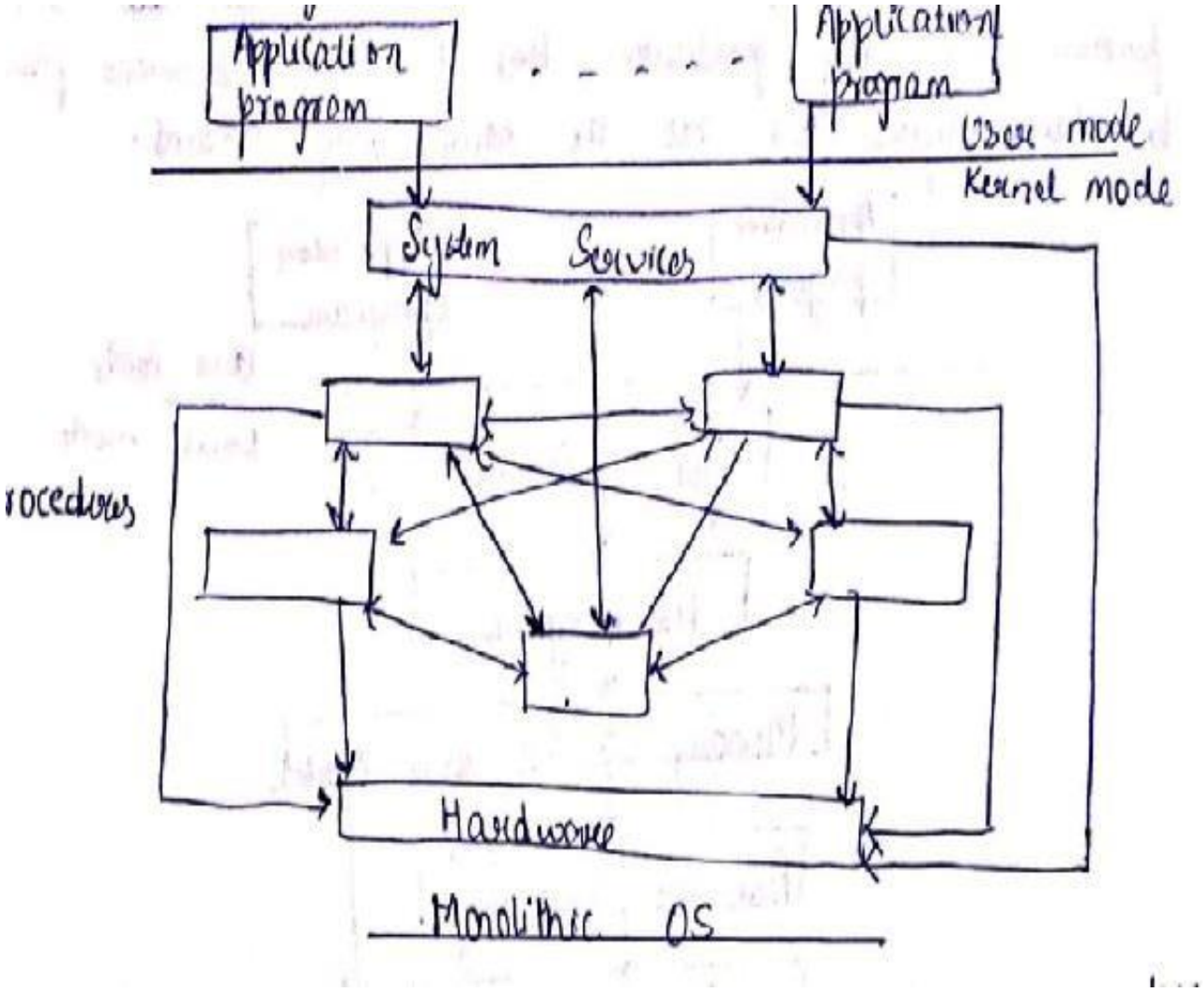
Example- VAX VMS, Multics, UMX.

(i) Monolithic O.S structure

The components of monolithic O.S. are organized haphazardly and any module can call any other module without any reservation. Similar to the other OS, applications is monolithic OS are separated from the OS itself. i.e. OS code is runs in a privileged processor mode, with the access to system data and to the hardware, applications runs in a non-privileged processor mode called the user mode, with a limited set of interfaces available and with limited access to system data.

When a user mode program call a System Service, the processor traps the calls and then switches the calling thread Kernel mode. Completion of the system service, switches the thread back to the user made, by the O.S. and allow the caller to continue.

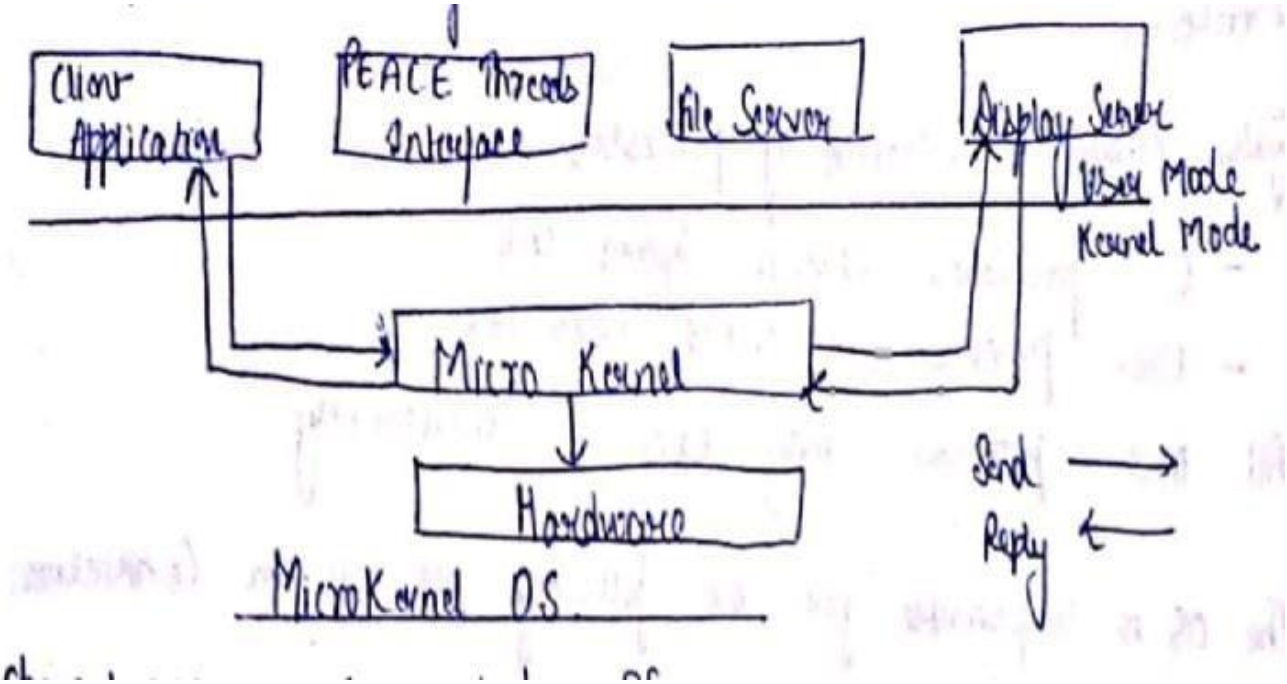
The monolithic structure does not enforce data hiding in the O.S. It delivers better application performance, but extending such a system can difficult work. Because modifying a procedure can introduce bugs in seeming unrelated parts of the system.



Example:-CP/M and MS-DOS.

(j) Micro Kernel and Client Server O.S.

The advent of the new concepts in OS design, micro Kernel, is aimed at migrating traditional services of an OS out of the monolithic Kernel into the user level process. The idea is the divide the OS into several processes, each of which implements a single sets services. Each server runs in the user mode, provide services to the requested client. The client, which can be either another a components or application programs, request a services by sending msg. to the server. An OS kernel (or micro Kernel) running in the Kernel mode delivers the msg. to the appropriate server, the server performs the operation, and microkernel deliver the result to the client in another msg.



Characteristics:-

- Simplified base OS,
- Improved reliability,
- Traditional services of OS have become peripheral,
- Message passing facilities,
- Leads to a distributed computing model,
- Foundations for modular and portable extensions

Example:- C-DAC Paras, Windows NT195, Mach, QNX, Chows

IV. System Components

(a) Process Management

A process can be through of as program in execution. A process will need certain resources -such as CPU time, memory file and I/O devices to accomplish its task. These resources are allocated to the process either when it is created or white while is it executing.

A program by itself is not a process. a program is a passive entity, such as a file containing list of instruction Stored on disk, whereas in an active entity, with a program counter specifying the next instruction to execute and a set of associated resources.

- **System Consist of collection of processes-**
- OS processes execute system code.
- User processes execute user code.

All those processes may execute concurrently.

The OS is responsible for the following activities in connection with process management..

- The creation and detection of both user and system processes.
- The suspension and resumption of process.
- The provision of mechanism for process in synchronization.
- The provision of mechanism for process communication.
- The provision of mechanism for deadlock handling.

(b) Main memory management

Main memory is control to the operation of a modern computer system. Main memory is a large array of words or bytes. Main memory is a repository of quickly accessible data share by the CPU and I/O devices. The central processor reads instructions from main memory during the instruction fetch cycle. The main memory is the only large storage device that the CPU is able to address and access directly.

Selection of memory management scheme for specific system depends on many factors, specifically on the hardware design of a system.

The OS is responsible for the following activities in connection with memory mgmt.-

- Keep track of which parts of memory are currently being used and by whom.
- Decide which processes are to be loaded into memory when memory is space becomes available.
- Allocate and deallocate memory space as needed.

(c) File management

Computers can store information on several different type of physical media, such as Magnetic tape, disk etc. for convenient use of the computer system, the OS provides a uniform logical view of information storage. A file is a collection of related information defined by its creator.

The OS is responsible for the following activities in connection with file management.

- The creation and deletion of file.
- The creation and deletion of directories.
- The support of primitives for manipulating files and directories.
- The mapping of files onto secondary storage.
- The backup of files and stable storage media.

(d) Secondary Storage Management

The main purpose of computer system is execute programs. These programs with the data they access, must be in main memory during execution because main memory is too small to accommodate all data and programs and its data is lost when power is lost the computer system must provide secondary storage to backup main memory.

The operating system is responsible for the following activities in connection with disk management-

- Free Space Management.
- Storage Allocation.
- Disk Scheduling.

(e) User – Interface:-

The OS need to provide an interface to communicate with the user. This could be command line Interface or a graphical user Interface.

A CLI is a method of interacting with an OS or software using a command line interpreter. This command line interpreter may be a text terminal, terminal emulator, or remote shell client.

A GUI is a type of user Interface which allows people to interact with a computer and computer controlled devices which employ graphical icons.

(f) I/O System Management:-

One of the purpose of an OS is to hide the peculiarities of specific hardware devices from the user, For Example, in UNIX, the peculiarities of I/O devices are hidden from the buk of an OS itself by the I/O subsystem. The I/O subsystem consist of-

- A memory mgmt. component including buffering, caching and spooling.
- A general device – driver interface.
- Drivers for specific hardware devices.

(g) **Networking:-**

A distributed system is a collection of processors that do not share memory, peripheral devices or a clock. Instead each processor has its own local memory and clock, and the processor communication with one another through various communication lines such as high speed buses or telephone lines. The processor in a distributing system vary in size and function. They may include small microprocessor, workstations, mini computers and large general purpose computer systems. The processor in the system are connected through a communication network which can be configured in a number of different ways.

(k) Protection system

If a computer system has multiple users and allows the concurrent execution of multiple processors than the various processes must be protected from one another activities. Protection refers to mechanism for controlling the access of programs processes or users to resources defined by computer system. The mechanism must be provide a means for specification of the control, to be imposed, together with a means of enforcement.

(l) Command Interpreter system

One of the most important system programs for an OS is the command interpreter, which is the interface between the users and the OS.

Many commands are given to the OS by control statements. When a new job is started in a batch systems, or when a user logs on to a time-shared systems, a program that reads and interprets control statements is executed automation. This program is sometimes called the control- card interpreter or the command - line interpreter.

V. Operating Systems Services

An Operating System provides an environment for execution of program. The operating system provided certain services to program and to the user of those programs. The operating system services are provided for the convenience of the programmer, to make the programming easier.

(a) Program execution

The system must be able to load a program into memory and to run it. The program must be able to end its execution, either normally or abnormally (indicating error).

(b) I/O operation

Running program may require input output. This input output may involve a file or an input output device. For specific devices, special functions may be desired. For efficiency and protection, users usual cannot control input output device directly, the operating system must provide some means to do input output.

(c) File system manipulation

The file system is of particular interest. It should be obvious the program need to read and files. They also need to create and delete files by name.

(d) Communication

There are many circumstances in which one process need to exchange information with another process. They are two ways in which such communication can occur. The first take place between processes executing on the same computer. The second takes place between processes execute on different computer system that are tied together by computer network. Communication may be implemented via shared memory or buy the technique of message passing.

(e) Errors detection

Error may occur in the CPU and memory hardware in output input devices, or in the user program. For each type of error, the operating system should take the appropriate action to ensure correct and consistent computing.

(f) Resource allocation

And there are multiple users or multiple jobs running at the same time, resources must be allocated to each of them. Many different type of resources and managed by the operating system.

(g) Accounting

We want to keep track of which users use how much and what kind of computer resources.

(h) Protection

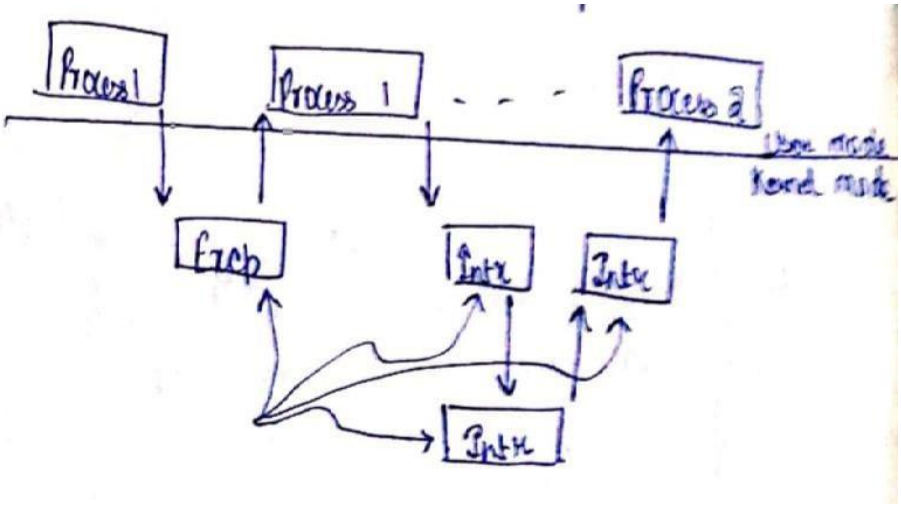
The owner of information stored in a multi user computer system may work to control its use. When several disjoint process execute concurrently, it should not be possible for one process to interface with the others, or with of the operating system itself. Protection involves in ensuring that all access to system resources is controlled.

VI. Re-Entrant Kernels

A Re-entrant Kernel enables process to give away the CPU while in Kernel mode, not hindering other processes from also entering Kernel mode.

A typical use is input output wait. The processes wants to read a file and calls Kernel function for this, inside the kernel function, the disk controller is asked for the data. Getting the data will take some time and the function is block during that time.

With and Re-Entrant Kernel, the scheduler will assign CPU to another process until and interrupt from the disk controller indicates that the data is available and our thread Can be assumed. This process can still access input output, like other input.



- Several process may be in Kernel mode at the same time.
- A Re-entrant Kernel is able to suspend the current running process even if it is in the Kernel mode.

VII. System Protection:

To improve system utilization, the operating system share system resources among several programs simultaneously.

This sharing created both utilization and increased problems. When the system was run without sharing, an error in a program could cause problems for only the one program that was running. With sharing, many processes could be adversely affected by a bug in one program.

Errors can occur in a multiprogramming system, where one erroneous program might modify the program or data of another program.

A properly designed operating system must ensure that an incorrect program can't cause other programs to execute correctly.

1). Dual Mode Operation: There are two modes of operations that are used for providing the protection

a) User Mode
b) Monitor mode or Kernel mode or Privileged mode

A bit is added to the computer hardware monitor (0) or user (1).

At the system boot time, system starts in monitor mode. The operating s/m is then loaded, and starts user process in user mode. Whenever a trap or interrupt occurs, the hardware switches from user mode to monitor mode.

The dual mode of operation provides us with the means for protecting the operating system from errant users, and errant users from one another. We accomplish this protection by designating some of the machine instructions that may cause harm as privileged instructions. The hardware allows privileged instructions to be executed in only monitor mode. If an attempt is made to execute a privileged instruction in the user mode, the hardware does not execute the instruction, but rather treats the instruction as illegal and traps to the operating system.

2). I/O Protection: A user may disrupt the normal operation of the system by issuing illegal I/O instructions, by accessing memory locations within the operating system itself, or by refusing to the relinquish the CPU.

To prevent a user from performing illegal I/O , we define all I/O instructions to be privileged instructions. Thus, users can't issue I/O instructions directly; they must do it through the OS.

3). Memory Protection: For providing memory protection, two registers are used-

a) Base Register- holds the smallest legal physical memory address.

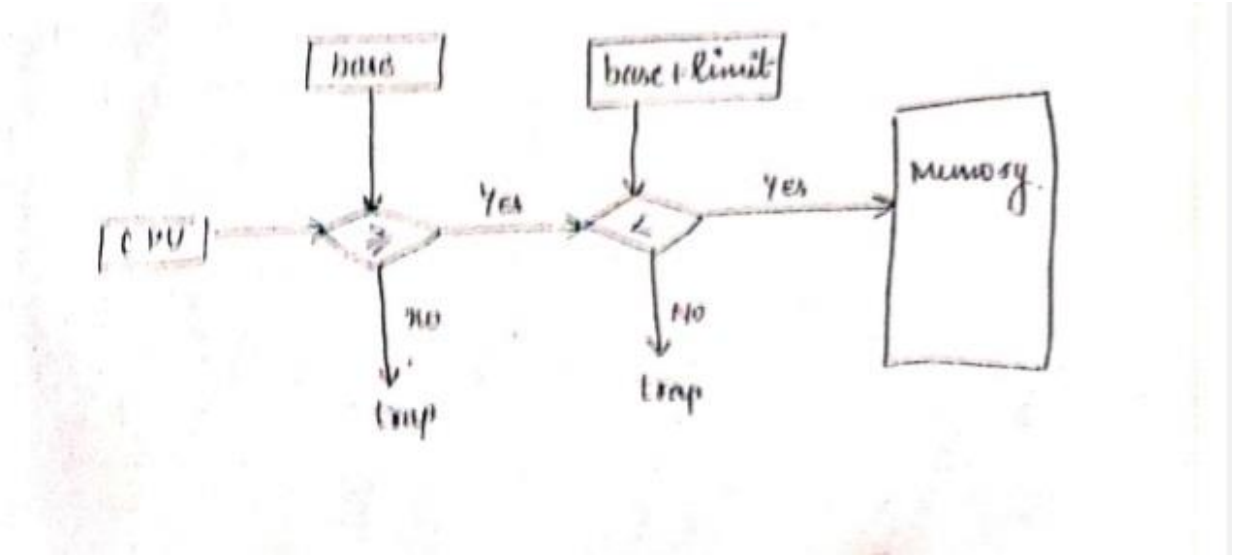
b) Limit Register- contains the size of the range.

For Example- if the base register holds 300400 and limit register is 120900, then the program can loyalty access all addresses from 300400 through 420940 inclusive.

This protection is accomplished by the CPU hardware compare every, address generated an user mode with the register. They attempt by a program executing in user mode to access monitor memory or other's user memory results in a trap to the monitor.

This scheme prevents the user program from modifying the code of either OS or other users.

The base and limit register can be loaded only by OS, which user a privileged instructions.



4). CPU Protection: There is a need to prevent a user program from getting stuck in an infinite loop, and never returning control to the operating system. To accomplish this goal, timer is used.

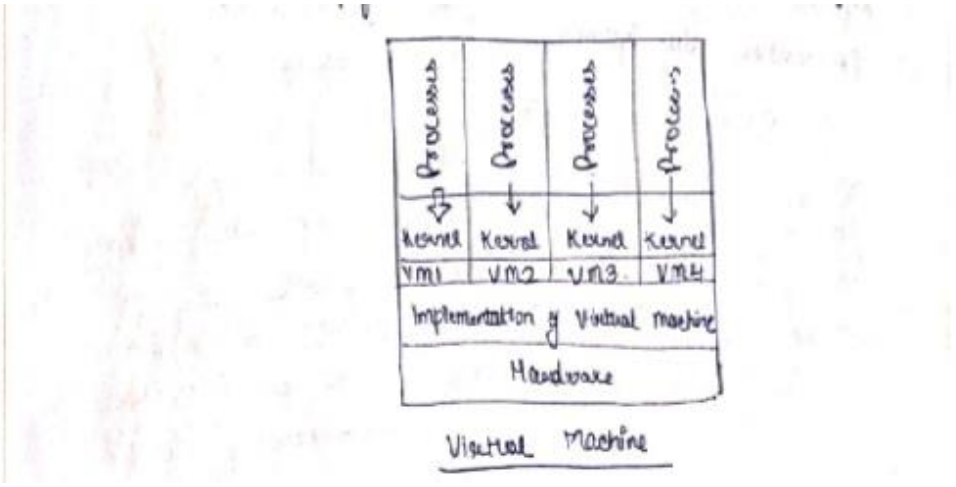
A timer can be set to interrupt the computer after a specified period. The period may be fixed as variable. The operating system sets the counter. Every time that clock ticks, the counter is decremented. When the counter reaches 0, an interrupt occur. If the timer interrupts, controls transfers automatically to the operating system.

A more common use of timer is to implement time sharing. In the most straight forward case, the timer could be ser to interrupt every N milliseconds, where N is the time slice that each user is allowed to execute before the next user gets control of the CPU.

VIII: Virtual Machine Architecture

A design concept in which the programming model is implemented by the OS rather than by underlying physical hardware. That is the OS provides a simulation of H/W for the programmers. The virtual machine OS for the 1 BM system is the best example of the virtual machine concept.

Multiple interactive sessions are implemented by a set of virtual machines one for each user, each user directs the virtual machine to perform different commands. These commands are the executed on the physical machine in a multiprogramming environment.



OS operates in 2 modes- user & Kernel mode.
The virtual m/c s/w can run in monitor mode.
But virtual m/c itself executes in only user mode.
Virtual user mode and a virtual monitor mode both are in the physical user mode.

Some actions that require the transfer from user mode to monitor mode in real machines, must also cause a transfer from virtual user mode to virtual monitor mode on a virtual machine.

Components of Virtual M/C-

1). Control program (CP) – Controls the physical machine.

2). Conversational Monitor System (CMS) – Controls the virtual machine.

Advantages-

1).Each virtual machine is completely isolated from all other virtual machines, so we have no security problems so the various system resources are completely protected.

2). It allows system development to be done without disrupting norm.

Example- LINUX

Disadvantages-

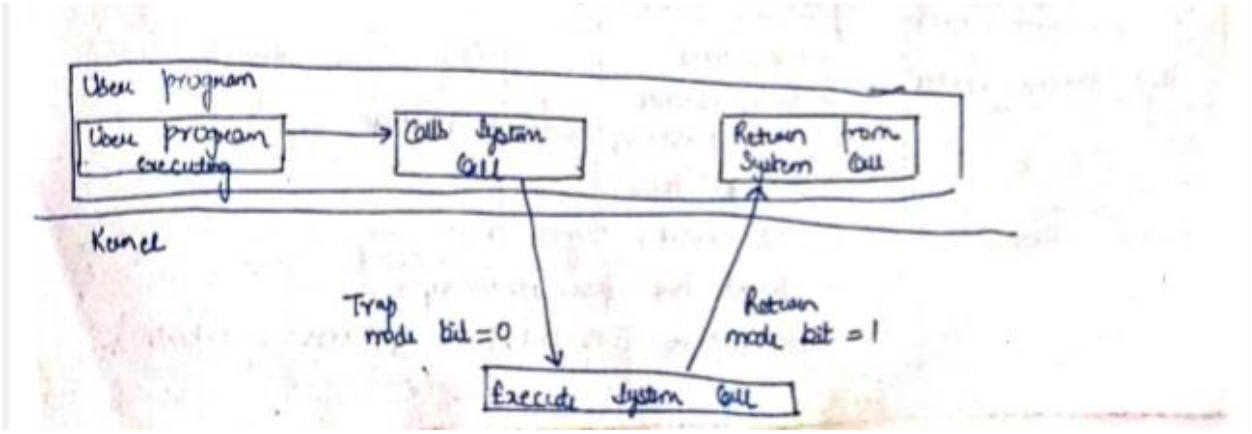
1). Difficult to implement.

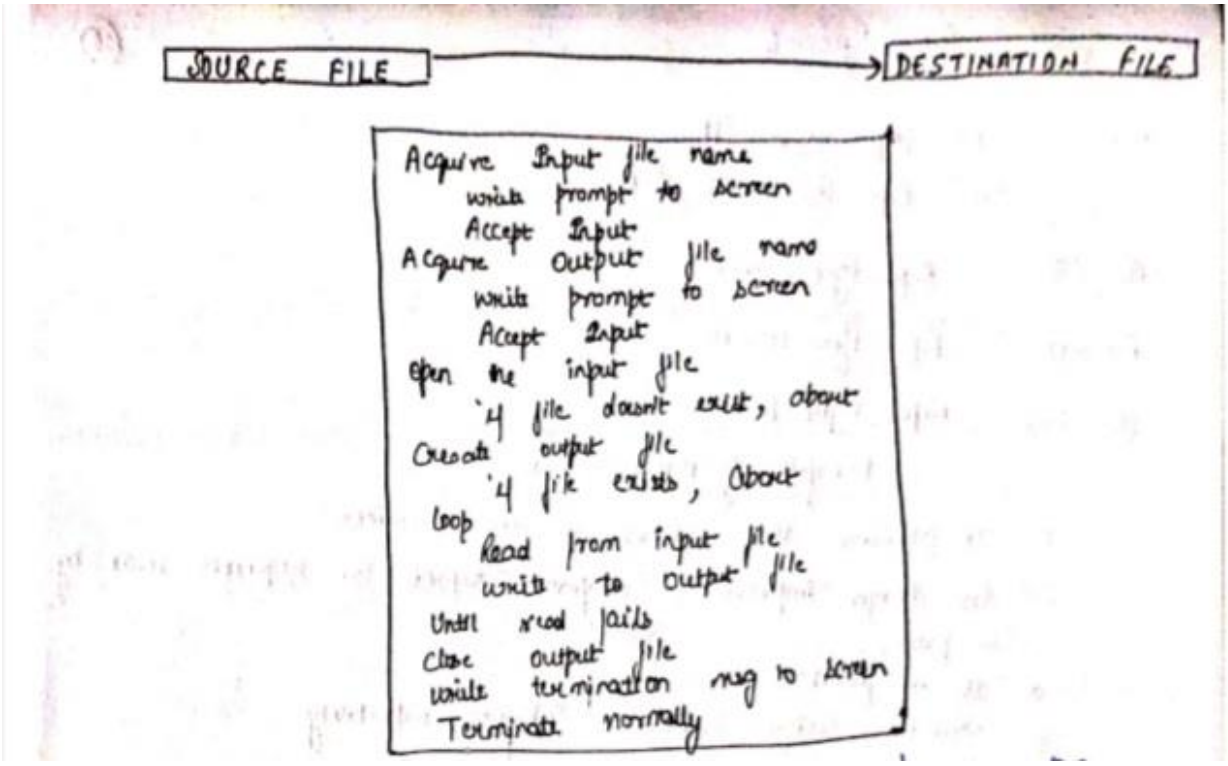
2). It involves disk system.

IX: System Calls

System Calls provide the interface between a process and the operating system. These calls are generally available as assembly – language instructions, and are usually listed in the manuals used by the assembly-language programmers.

Example- Writing a simple program to read data from one file and to copy them to another file.





Three general methods to pass parameters to the OS –

(a) Simplest approach to pass the parameters in Registers.
(b) In some cases, there may be more parameters than registers. In these cases, parameters are generally stored in a block, and the address of the block is passed as a parameters in registers.
(c) Parameters can also be pushed, or popped or the stack by the OS.

Categories of System Calls-

(a) Process Control-

- end , about
- Load, Execute
- Create process, terminate process
- Wait for time
- Wait event, Single Event
- Allocate and free memory
- Get process attribute, set process attribute.

(b) File Manipulation:

- Create File, Delete File
- Open, Close
- Read, Write, Reposition
- Get File attribute, set file attribute

(c) Device Manipulation:

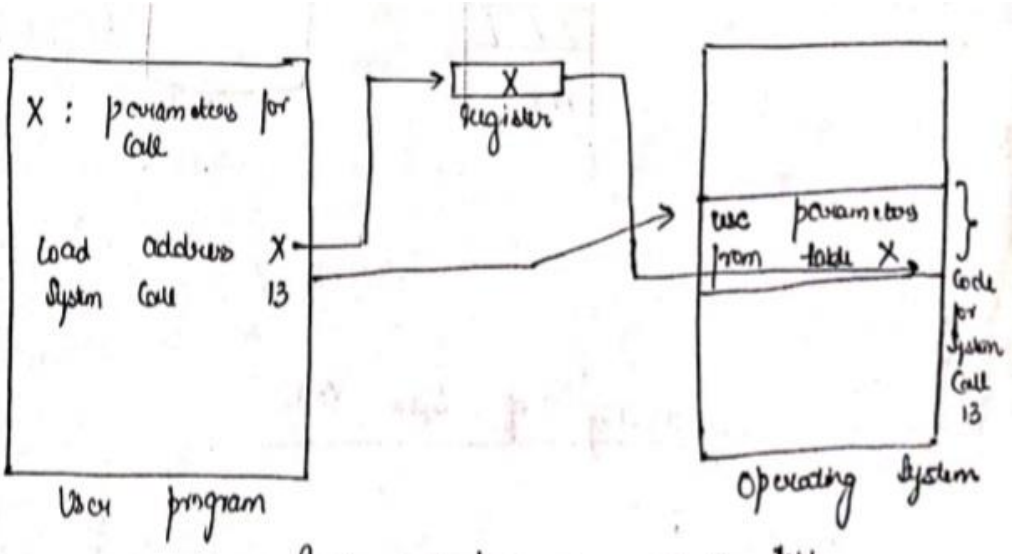
- request device, release device
- Read, write, reposition
- Get device attribute, set device attribute
- Logically attach or detach devices.

(d) Information Maintenance:

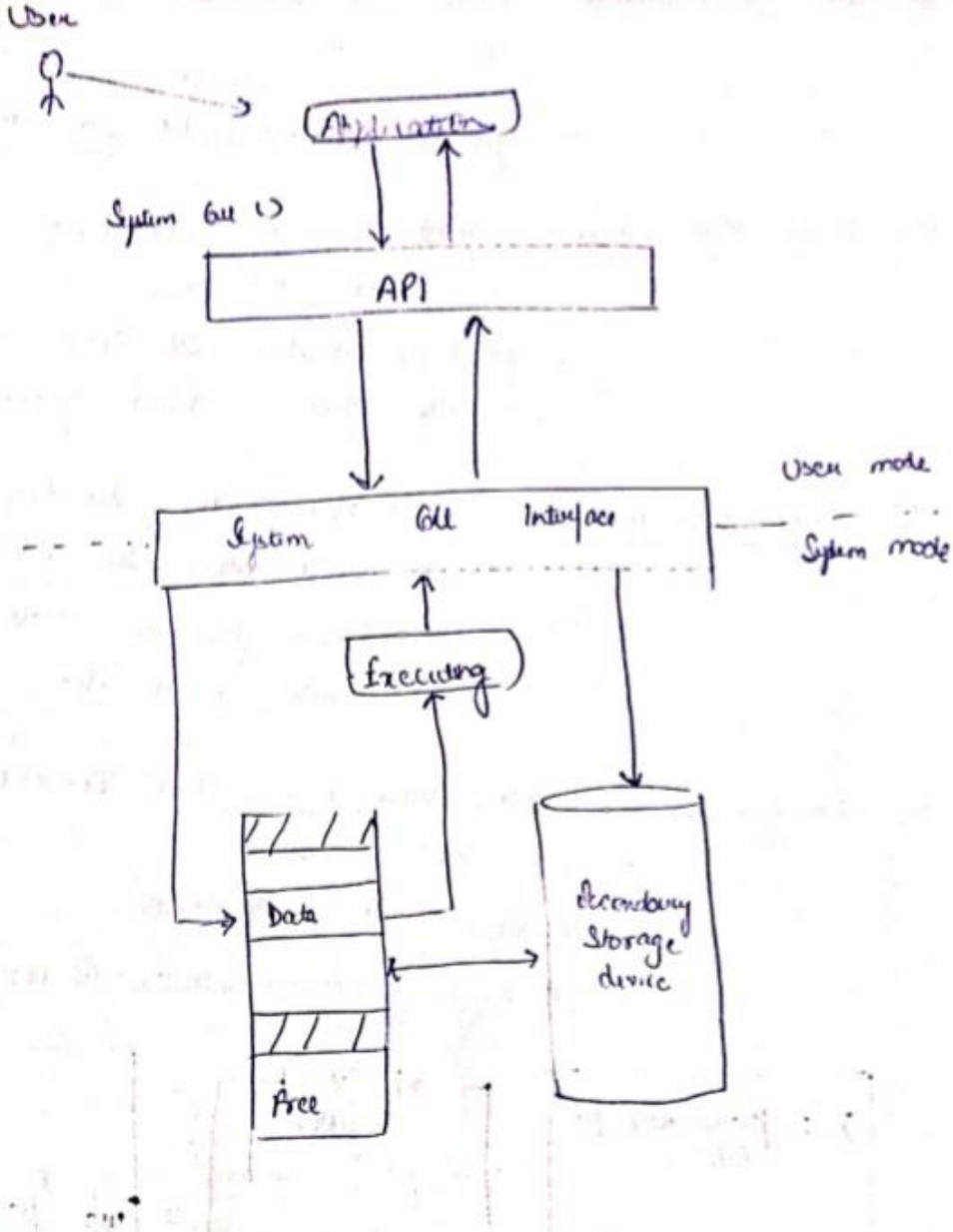
- get time or date, set time or date
- Get system data, set system data
- Get process, file or device attributes
- Set process, file or device attributes.

(e) Communication:

- create, delete communication connections
- Send, receive messages.
- Transfer status information.
- Attach or detach remote devices.



Passing of parameters as a Table



Working of System Call