OS (Introduction and OS Structure)

★★Computer System Components

A computer system can be divided roughly into four components: the *hardware,* the *operating system,* the *application programs,* and the *users* (Figure 1.1).

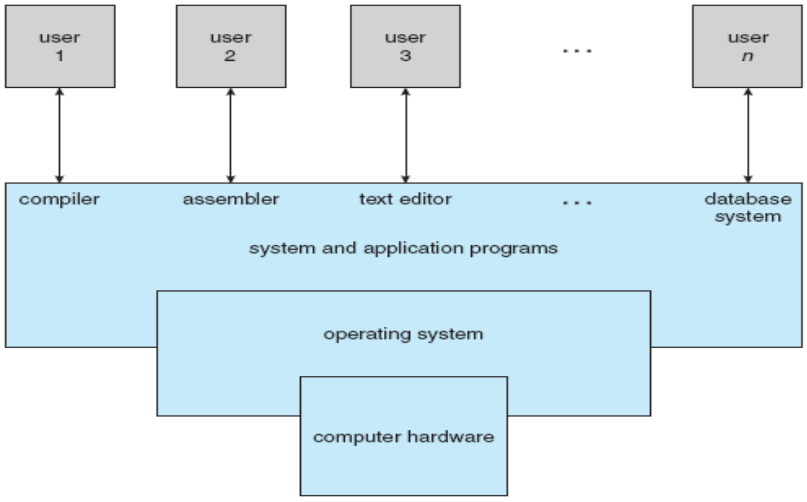

Figure 1.1 Abstract view of the components of a computer system.

- **Hardware**— provides the basic computing resources for the system. [**central processing unit (CPU)**, **memory**, **input/output (I/O) devices]**
- **Application programs**— define the ways in which these resources are used to solve users' computing problems [word processors, spreadsheets, compilers, and Web browsers]
- **Operating system** — controls the hardware and coordinates its use among the various application programs for the various users. An operating system is similar to a *government*. Like a government, it performs no useful function by itself. It simply provides an *environment* within which other programs can do useful work.
- **Users** — using computer system [people, machine, other computer]

★★Operating System

An operating system (OS) is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.

Fig – Same as above question

★★Objective of OS

- To hide details of hardware by creating abstraction.
- To allocate resources to processes.
- Provide a pleasant and effective user interface.

★★Three main purposes of an operating system

- To provide an environment for a computer user to execute programs on computer hardware in a convenient and efficient manner.
- To allocate the separate resources of the computer as needed to solve the problem given. The allocation process should be as fair and efficient as possible.
- As a control program it serves two major functions: (1) supervision of the execution of user programs to prevent errors and improper use of the computer, and (2) management of the operation and control of I/O devices.

★★OS Goals

- ✓ The primary goal of an operating system is to make the computer system *convenient to use*.
- ✓ A secondary goal is to use the computer hardware in an efficient manner.

★★OS challenges

•Reliability

- ✓ Does the system do what it was designed to do?

•Availability

- ✓ What portion of the time is the system working?
- ✓ Mean Time To Failure (MTTF), Mean Time to Repair

•Security

- ✓ Can the system be compromised by an attacker?

•Privacy

- ✓ Data is accessible only to authorized users

•Performance

- ✓ Latency/response time - How long does an operation take to complete?
- ✓ Throughput - How many operations can be done per unit of time?
- ✓ Overhead - How much extra work is done by the OS?
- ✓ Fairness - How equal is the performance received by different users?

•Portability

- ✓ For programs: Application programming interface (API)
- ✓ For the kernel: Hardware abstraction layer

★★Functions Of Operating System [Or, Operating System components. Answer: 1st four points]

Memory Management

Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address. Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must in the main memory.

An Operating System does the following activities for memory management:

- Keeps tracks of primary memory, i.e., what part of it are in use by whom, what parts are not in use.
- Allocates the memory when a process requests it to do so.
- De-allocates the memory when a process no longer needs it or has been terminated.

Processor Management

In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called **process scheduling**.

An Operating System does the following activities for processor management:

- Keeps tracks of processor and status of process. The program responsible for this task is known as **traffic controller**.
- Allocates the processor (CPU) to a process.
- De-allocates processor when a process is no longer required.

Device Management

An Operating System manages device communication via their respective drivers. It does the following activities for device management:

- Keeps tracks of all devices. The program responsible for this task is known as the **I/O controller**.
- Allocates the device in the most efficient way.
- De-allocates devices.
- Decides which process gets the device when and for how much time.

File Management

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions.

An Operating System does the following activities for file management:

- Keeps track of information, location, uses, status etc. The collective facilities are often known as **file system**.
- Allocates the resources.
- De-allocates the resources.
- Decides who gets the resources.

Security

By means of password and similar other techniques, it prevents unauthorized access to programs and data.

Job accounting

Keeping track of time and resources used by various jobs and users.

★★Explore Operating Systems

Two viewpoints to explore OS: the **user** and the **system.**

(1) User view:

The user's view of the computer varies according to the interface being used.

●PC

- ✓ Most computer users sit in front of a PC, consisting of a monitor, keyboard, mouse, and system unit. Such a system is designed for one user to monopolize its resources.
- ✓ The goal is to maximize the work (or play) that the user is performing. Some attention paid to performance and none paid to resource utilization
- ✓ Designed issue: ease of use

●Mainframe or a **minicomputer**

- ✓ In other cases, a user sits at a terminal connected to a **mainframe** or a **minicomputer**. Other users are accessing the same computer through other terminals.
- ✓ These users share resources and may exchange information.
- ✓ Designed issue: maximize resource utilization

●Workstations

- ✓ In still other cases, users sit at **workstations** connected to networks of other workstations and **servers**.
- ✓ These users have dedicated resources at their disposal, but they also share resources such as networking and servers—file, compute, and print servers.
- ✓ Designed issue: compromise between individual usability and resource utilization.

●Handheld computers

- ✓ Recently, many varieties of handheld computers have come into fashion. Most of these devices are standalone units for individual users. (Ex- Barcode Scanner)
- ✓ Some are connected to networks, either directly by wire or (more often) through wireless modems and networking.
- ✓ Because of power, speed, and interface limitations, they perform relatively few remote operations.
- ✓ Designed issue: individual usability, but performance per unit of battery life is important as well.

●Embedded computers

- ✓ Little or no user view.
- ✓ May have numeric keypads and may turn indicator lights on or off to show status
- ✓ Designed issue: primarily to run without user intervention. (Ex- Weight Machine)

(2)System View

●Resource allocator

- ✓ From the computer's point of view, the operating system is the program most intimately involved with the hardware. In this context, we can view an operating system as a **resource allocator**.
- ✓ A computer system has many resources that may be required to solve a problem: CPU time, memory space, file-storage space, I/O devices, and so on.
- ✓ The operating system acts as the manager of these resources.
- ✓ Facing numerous and possibly conflicting requests for resources, the operating system must decide how to allocate them to specific programs and users so that it can operate the computer system efficiently and fairly.

●Control program

- ✓ An operating system is a control program.
- ✓ A **control program** manages the execution of user programs to prevent errors and improper use of the computer.
- ✓ It is especially concerned with the operation and control of I/O devices.

★★Operating System History [Evolution of OS] [Types of OS]

Operating systems and computer architecture have influenced each other a great deal. To facilitate the use of the hardware, researchers developed operating systems. Users of the operating systems then proposed changes in hardware design to simplify them. In this short historical review, we will see how identification of operating-system problems led to the introduction of new hardware features.

1. Mainframe Systems

Mainframe computer systems were the first computers used to tackle many commercial and scientific applications. In this section, we trace the growth of mainframe systems from simple batch systems, where the computer runs one and only one-application, to time-shared systems, which allow for user interaction with the computer system.

1.1 Batch Systems

- Early computers were physically enormous machines run from a console.
- The common input devices were card readers and tape drives.
- The common output devices were line printers, tape drives, and card punches.
- The user did not interact directly with the computer systems. Rather, the user prepared a job - which consisted of the program, the data, and some control information about the nature of the job (control cards)-and submitted it to the computer operator. The job was usually in the form of punch cards.
- At some later time (after minutes, hours, or days), the output appeared. The output consisted of the result of the program, as well as a dump of the final memory and register contents for debugging.
- The operating system in these early computers was fairly simple. Its major task was to transfer control automatically from one job to the next. The operating system was always resident in memory (Figure 1.2).

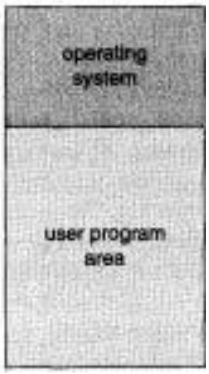

Figure **1.2** Memory layouts for a simple batch system.

1.2 Multiprogrammed Systems

- The most important aspect of job scheduling is the ability to multiprogram. Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.
- The idea is as follows: The operating system keeps several jobs in memory simultaneously (Figure 1.3). This set of jobs is a subset of the jobs kept in the job pool.

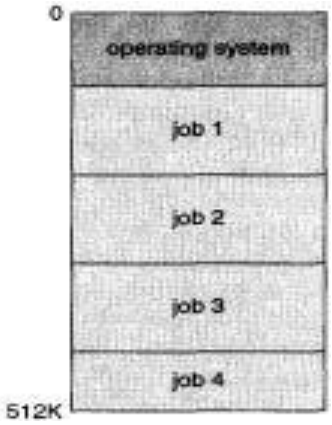

Figure 1.3 Memory layout for a multiprogramming system.

- Operating system picks and begins to execute one of the jobs in the memory. Eventually, the job may have to wait for some task, such as an I/O operation, to complete.
- The operating system simply switches to, and executes, another job. When *that* job needs to wait, the CPU is switched to *another* job, and so on.
- Eventually, the first job finishes waiting and gets the CPU back. As long as at least one job needs to execute, the CPU is never idle.
- If several jobs are ready to be brought into memory, and if there is not enough room for all of them, then the system must choose among them. Making this decision is ***job scheduling.***
- If several jobs are ready to run at the same time, the system must choose among them. Making this decision is CPU scheduling.

1.3 Time-Sharing Systems

- Time sharing (or multitasking) is a logical extension of multiprogramming. The CPU executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while it is running.
- A time-shared operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared computer.

2. Desktop Systems

- Single user systems, portable.
- I/O devices - keyboards, mice, display screens, small printers.
- Single user systems may not need advanced CPU utilization or protection features.
- Advantages: user convenience, responsiveness, ubiquitous

3. Multiprocessor Systems

●Such systems have more than one processor in close communication, sharing the computer bus, the clock, and sometimes memory and peripheral devices. Multiprocessor systems have three main advantages.

- ✓ Increased throughput.
- ✓ Economy of scale.
- ✓ Increased reliability.

●The most common multiple-processor systems now use symmetric multiprocessing (SMP). Some systems use asymmetric multiprocessing

4. Distributed Systems

Distributed systems use multiple central processors to serve multiple real-time applications and multiple users. Data processing jobs are distributed among the processors accordingly. The advantages of distributed systems are as follows:

- ➢ With resource sharing facility, a user at one site may be able to use the resources available at another.
- ➢ Speedup the exchange of data with one another via electronic mail.
- ➢ If one site fails in a distributed system, the remaining sites can potentially continue operating.
- ➢ Better service to the customers.
- ➢ Reduction of the load on the host computer and Reduction of delays in data processing.

5. Real-Time Systems

A real-time system is defined as a data processing system in which the time interval required to process and respond to inputs is so small that it controls the environment. The time taken by the system to respond to an input and display of required updated information is termed as the **response time**. There are two types of real-time operating systems.

Hard real-time systems: Hard real-time systems guarantee that critical tasks complete on time. In hard real-time systems, secondary storage is limited or missing and the data is stored in ROM. In these systems, virtual memory is almost never found. Examples of hard real-time systems: **airplane sensor and autopilot systems.**

Soft real-time systems: Soft real-time systems are less restrictive. A critical real-time task gets priority over other tasks and retains the priority until it completes. Soft real-time systems have limited utility than hard real-time systems. For example, multimedia, virtual reality, Advanced Scientific Projects like undersea exploration and planetary rovers, etc.

6. Clustered Systems

- ➢ Provide high reliability
- ➢ Asymmetric: One server runs the application while other servers standby.
- ➢ Symmetric: All N hosts are running the application

7. Handheld Systems

- Limited memory
- Slow processors
- Small display screens

★★Major advantages and disadvantages of multiprocessor system

Advantages:

- ✓ Increased throughput.
- ✓ Economy of scale.
- ✓ Increased reliability.

Disadvantages:

- ✓ It's more complex
- ✓ It requires context switching which slightly impacts performance.

★★Operating System Services

Following are a few common services provided by an operating system:

- Program execution
- I/O operations
- File System manipulation
- Communication
- Error Handling

Program Execution

Following are the major activities of an operating system with respect to program management:

- Loads a program into memory
- Executes the program
- Handles program's execution
- Provides a mechanism for process synchronization

I/O Operation

- I/O operation means read or write operation with any file or any specific I/O device.
- Operating system provides the access to the required I/O device when required.

File System Manipulation

A file represents a collection of related information. Computers can store files on the disk (secondary storage), for long-term storage purpose. Following are the major activities of an operating system with respect to file management:

- Program needs to read a file or write a file.
- The operating system gives the permission to the program for operation on file.
- 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.

Communication: Following are the major activities of an operating system with respect to communication:

- Two processes often require data to be transferred between them.
- Both the processes can be on one computer or on different computers, but are connected through a computer network.
- Communication may be implemented by two methods, either by Shared Memory or by Message Passing.

Error Handling

Errors can occur anytime and anywhere. An error may occur in CPU, in I/O devices or in the memory hardware. 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.

★★Computing Environments

Development of operating systems from the first hands-on systems through multiprogrammed and time-shared systems to PCs and handheld computers. We can give a brief overview of how such systems are used in a variety of computing environment settings.

- **Traditional Computing:** Typical office environment uses traditional computing. Normal PC is used in traditional computing environment. Network computers are essential terminals that understand web based computing. In domestic application most of the user had a single computer with internet connection. Cost of accessing internet is high.

- **Web Based Computing:** It has increased the emphasis on Networking. Web based computing uses PC, handheld PDA & cell phones. One of the feature of this type is load balancing. In load balancing, N/w connection is distributed among a pool of similar servers.

- **Embedded computing:** It uses real time OS. Application of embedded computing is car engines, manufacturing robots, microwave ovens. This type of system provides limited features.

OS (Operating System Structures)

★★A modern computer system

A modern, general-purpose computer system consists of a **CPU** and a number of device controllers that are connected through a common bus that provides access to shared memory (Figure 2.1). Each device controller is in charge of a specific type of device (for example, disk drives, audio devices, and video displays). The CPU and the device controllers can execute concurrently, competing for memory cycles. To ensure orderly access to the shared memory, a memory controller is provided whose function is to synchronize access to the memory.

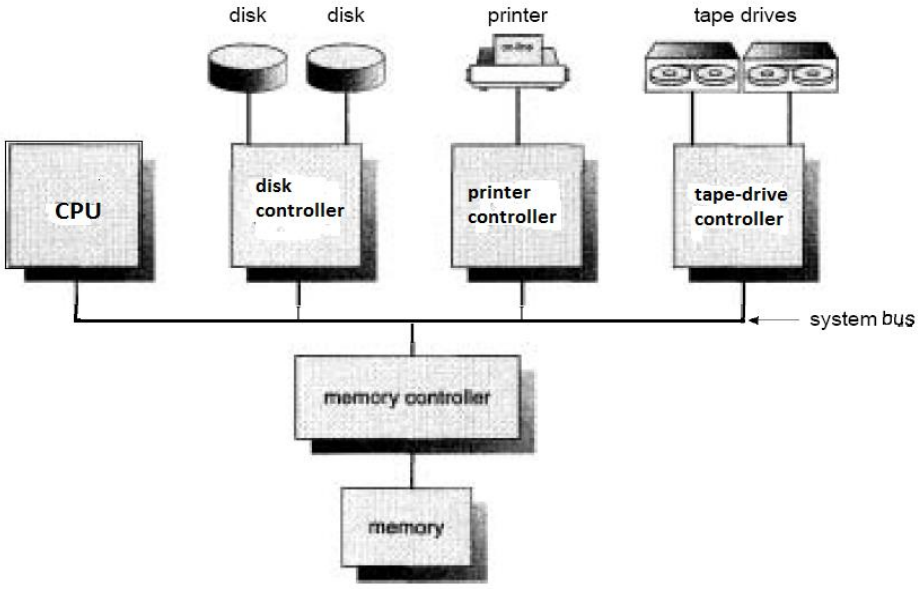

Figure 2.1 A modern computer system.

★★Bootstrap program

For a computer to start running-for instance, when it is powered up or rebooted - it needs to have an initial program to run. The bootstrap program must know how to load the operating system and to start executing that system.

★★Interrupts

A method by which other events can cause an interruption of the CPU's normal execution. Four general classes of interrupts are :

- Program, trap instructions, page faults etc.
- Timer
- I/O devices and
- Hardware failure.

★★Interrupt cycle

When an interrupt occurs a register in the CPU will be updated. When the CPU finishes the current execute cycle, and when interrupts are enabled, it will examine the register. If the register indicates that an interrupt has occurred and is enabled the interrupt cycle will begin, otherwise it will be bypassed [disabled]. The CPU follows the simple program outlined in the following diagram.

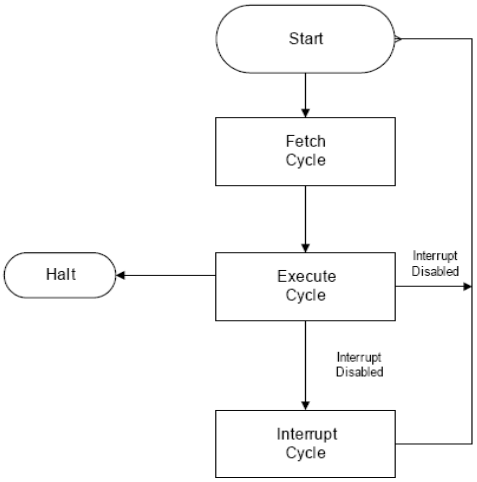

Fig.2.1 : Interrupt cycle with interrupts.

Interrupts are disabled when the operating system wishes to execute some code that must not be interrupted. Examples include interrupt handlers.

★★Simple Interrupt Processing

Steps for processing interrupts are shown below where steps 1 to 5 is done by hardware and from 6 to 9 is done by software

1. Interrupt occurs

2. Processor finishes current instruction

3. Processor signals acknowledgment of interrupt

4. Processor pushes program status word (Program Status Word) and program counter (Program Counter) onto stack

5. Processor loads **new** (নতুন যে Program নিয়ে কাজ করবে তার জন্য) Program Counter value based on interrupt

6. Save remainder of process information

7. Process interrupt

8. Restore process state information

9. Restore Program Status Word and Program Counter.

★★I/O Structure

One of the main functions of an OS is to perform all the computer's I/O devices. It issues commands to the devices, catch interrupts and handle errors. It provides an interface between the devices and the rest of the system. We will discuss the I/O hardware and I/O Software.

☆☆I/O Hardware

The I/O hardware is classified as

1. I/O devices
2. Device controllers and
3. Direct memory access (DMA).

(1) I/O Devices

Normally all input and output operations in operating system are done through two types of devices; *block oriented devices* and *character oriented devices*. A *block oriented device* is one in which information is stored and transferred at some fixed block size (usually some multiple of 512 bytes), each one with its own address.

The *character oriented device* is one in which information is transferred via a character stream. It has no block structure. It is not addressable. For example, punched cards, terminals, printers, network interfaces, mouse etc.

(2) Device Controller [I/O controller]

- I/O units consist of mechanical and electronic components. The electronic component is called device controller. It is also called printed circuit card.
- The controller's job is to convert the serial bit stream into a block of bytes and perform any error correction necessary.
- Each controller has a few registers that are used for communicating with the CPU and these registers are part of the regular memory address space. This is a called memory mapped I/O.

I/O Controller	I/O Address
Keyboard	060 - 063
Hard disk	320 - 32F
Printer	378 - 37F
Floppy disk	3F0 - 3F7

Table 2.2 : Controllers and their addresses.

(3) Direct Memory Access

DMA (Direct Memory Access) unit is capable of transferring data straight from memory to I/O devices. DMA is used for Direct memory operation.

☆☆I/O Software

Considering software as a series of layers with lower ones concerned with hardware and upper ones concerned with the interfaces to the users. These goals can be achieved by structuring the I/O software in four layers.

- Interrupt handlers
- Device drivers
- Device independent OS software
- User level software

Interrupt Handlers

The interrupt handlers will call other processes to handle interrupts those should be hidden away, deep in the bowels of the OS.

Device Drivers

Device driver is a program that is used to control each device. All hardware components of the computer is called devices. We saw that each controller has one/ more device register used to give it commands. The device drivers issue these commands and check that they are carried out properly.

Device Independent I/O Software

Some of I/O software is device specific and others are device independent. The basic function of the device independent software is to perform the I/O functions that are common to all devices. The functions of device independent software are:

- Device naming.
- Device protection.
- Buffering.
- Error reporting.
- Uniform interfacing for the device drivers.

User Level I/O Software

Most of the I/O software is within the OS, a small portion of I/O software consist of library linked user programs and I/O system calls are normally made by library procedures.

★★Five major activities of OS in regard to process management

The OS is responsible for the following activities of the process management,

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

★★System call

A system call is a special type of function call that allows user programs to access the services provided by the operation system. System calls can be roughly grouped into three major categories:

- process **or** job control,
- device and file manipulation,
- And information maintenance.

The following summarizes the types of system calls normally provided by OS.

i). Process Control

- End, Abort
- Load
- Create Process, Terminate Process
- Get Process Attributes, Set Process Attributes

ii). File Manipulation

- Create File, Delete File
- Open, Close
- Read, Write, Reposition
- Get File Attributers, Set File Attributes.

iii). Device Manipulation

- Request Device, Release Device
- Get Device Attributes, Set Device Attributes.

iv). Information Maintenance

- Get System Data, Set System Data
- Get Processes, File or Device Attributes, Set Process, File Device Attributes.

★★System Program

An important aspect of a modern system is its collection of systems programs to solve common problems and provide a more convenient environment and execution.

Systems programs can be classified into several categories:

File Manipulation: These programs create, delete, copy, rename, print, dump, list, and generally manipulate files and directories.

Status Information: Some programs simply ask the operating system for the date, time, amount of available memory or disk space, number of users, or similar status information.

File Modification: Several text editors may be available to create and modify the content of files stored on disk or tape.

Programming Language Support: Compilers, assemblers, and interpreters for common programming languages (such as Fortran, Cobol, Pascal, Basic, and so on) are often provided with the operating system. But recently many of these programs are being priced and provided separately.

Program Loading and Execution: Once a program is assembled or compiled, it must be loaded into memory to be executed.

Application Program: Most operating systems come with programs which are useful to solve some particularly common problems, such as compiler-compilers, text formatters, plotting packages, database systems, statistical analysis packages, and so on.

The most important system program for an OS is its command *interpreter*. It is that program which is runs when a job initially starts or user first logs in to a time sharing system.

★★Monolithic System

- In **Monolithic System,** all function are implemented within a single module that contains all system routines or process and all operating system data structure.
- In monolithic systems, it is possible to have at least a little structure.
- Most CPUs have two modes; kernel mode, for the operating system, in which all instructions are allowed; and user mode, for user programs, in which I/O and certain other instructions are not allowed.

★★Client / Server or Micro-Kernel Approach

- A micro-kernel is a "new" way of structuring an operating system. Instead of providing all operating system services (as do most current kernels) a micro-kernel provides a much smaller subset.
- The *advantages* of this structure is as follows :
  - better way to write software
  - easier to distribute across many machines.
- The main *disadvantage* is the slow in speed.

★★Difference between monolithic and micro kernel system

- A monolithic operating system contains all the necessary code in the one kernel. This means that if any changes are made to the kernel the whole system must be rebooted for the changes to take effect.
- A micro-kernel operating system contains a much reduced set of code in the kernel of the operating system. Most of the services provided by the OS are moved out into separate user level processes.
- All communication within a micro-kernel is generally via message passing whereas a monolithic kernel relies on variables and local procedure calls.

Reference Book: Operating System Concepts 6e By Silberschatz, Galvin, Gagne