UNIT-I

1) Define OS and Explain about the Types of Operating System.

A) Definition: "**Operating System** is system software that works as an interface between a user and the computer hardware."

Batch operating system

The users of a batch operating system do not interact with the computer directly. Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group. The programmers leave their programs with the operator and the operator then sorts the programs with similar requirements into batches.

The problems with Batch Systems are as follows -

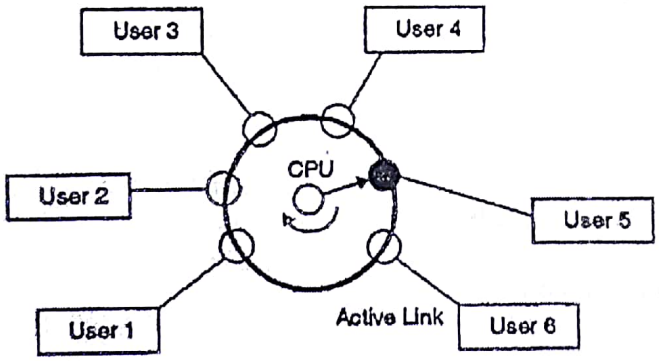
- Lack of interaction between the user and the job.
- CPU is often idle, because the speed of the mechanical I/O devices is slower than the CPU
- Difficult to provide the desired priority.

Time-sharing operating systems

Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time-sharing or multitasking is a logical extension of multiprogramming. Processor's time which is shared among multiple users simultaneously is termed as time-sharing.

The main aim of Time-Sharing Systems, the objective is to minimize response time.

Multiple jobs are executed by the CPU by switching between them, but the switches occur so frequently. Thus, the user can receive an immediate response. For example, in a transaction processing, the processor executes each user program in a short burst or quantum of computation. That is, if **n** users are present, then each user can get a time quantum. When the user submits the command, the response time is in few seconds at most.



In above figure the user 5 is active but user 1, user 2, user 3, and user 4 are in waiting state whereas user 6 is in ready status. As soon as the time slice of user 5 is completed, the control moves on to the next ready user i.e. user 6. In this state user 2, user 3, user 4, and user 5 are in waiting state and user 1 is in ready state. The process continues in the same way and so on.

The operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time. Computer systems that were designed primarily as batch systems have been modified to time-sharing systems.

Advantages of Timesharing operating systems are as follows –

- Provides the advantage of quick response.
- Avoids duplication of software.
- Reduces CPU idle time.

Disadvantages of Time-sharing operating systems are as follows –

- Problem of reliability.
- Question of security and integrity of user programs and data.
- Problem of data communication.

3 **Distributed operating System:**

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 processors communicate with one another through various communication lines (such as high-speed buses or telephone lines). These are referred as **loosely coupled systems** or distributed systems. Processors in a distributed system may vary in size and function. These processors are referred as sites, nodes, computers, and so on.

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.
- Reduction of delays in data processing.

4 **Network operating System:**

A Network Operating System runs on a server and provides the server the capability to manage data, users, groups, security, applications, and other networking functions. The primary purpose of the network operating system is to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks.

Examples of network operating systems include Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD.

The advantages of network operating systems are as follows –

- Centralized servers are highly stable.
- Security is server managed.
- Upgrades to new technologies and hardware can be easily integrated into the system.
- Remote access to servers is possible from different locations and types of systems.

The disadvantages of network operating systems are as follows –

- High cost of buying and running a server.

- Dependency on a central location for most operations.
- Regular maintenance and updates are required.

Real Time operating System:

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**. So in this method, the response time is very less as compared to online processing.

Real-time systems are used when there are rigid time requirements on the operation of a processor or the flow of data and real-time systems can be used as a control device in a dedicated application. A real-time operating system must have well-defined, fixed time constraints, otherwise the system will fail. For example, Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

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.

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.

2) Explain about the Operating System – Services.

A) An Operating System provides services to both the users and to the programs.

- It provides programs an environment to execute.
- It provides users the services to execute the programs in a convenient manner.

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

- User Interface
- Program execution
- I/O operations
- File System manipulation
- Communication
- Error Detection
- Resource Allocation
- Protection

User Interface

One is a **command-line interface(CLI)**, which uses text commands and a method for entering them. Most commonly, a **graphical user interface (GUI)** is used. Here, the interface is a window system with a pointing device to direct I/O, choose from menus, and make selections and a keyboard to enter text.

Program execution

Operating systems handle many kinds of activities from user programs to system programs like printer spooler, name servers, file server, etc. Each of these activities is encapsulated as a process.

A process includes the complete execution context (code to execute, data to manipulate, registers, OS resources in use). 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.
- Provides a mechanism for process communication.
- Provides a mechanism for deadlock handling.

I/O Operation

An I/O subsystem comprises of I/O devices and their corresponding driver software. Drivers hide the peculiarities of specific hardware devices from the users.

An Operating System manages the communication between user and device drivers.

- 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. Examples of storage media include magnetic tape, magnetic disk and optical disk drives like CD, DVD. Each of these media has its own properties like speed, capacity, data transfer rate and data access methods.

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

Communication

In case of distributed systems which are a collection of processors that do not share memory, peripheral devices, or a clock, the operating system manages communications between all the processes. Multiple processes communicate with one another through communication lines in the network.

The OS handles routing and connection strategies, and the problems of contention and security. 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.

Resource Management

In case of multi-user or multi-tasking environment, resources such as main memory, CPU cycles and files storage are to be allocated to each user or job. Following are the major activities of an operating system with respect to resource management –

- The OS manages all kinds of resources using schedulers.
- CPU scheduling algorithms are used for better utilization of CPU.

Protection

Considering a computer system having multiple users and concurrent execution of multiple processes, the various processes must be protected from each other's activities.

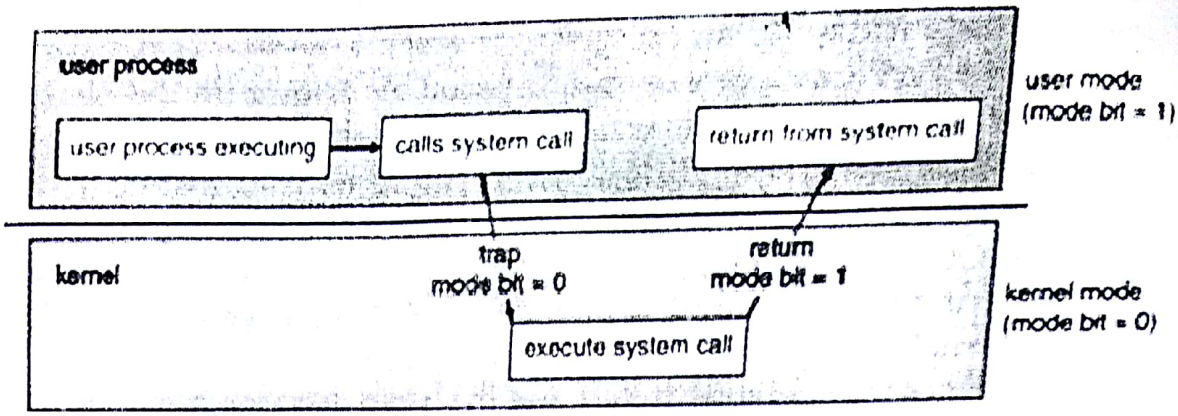
Protection refers to a mechanism or a way to control the access of programs, processes, or users to the resources defined by a computer system. Following are the major activities of an operating system with respect to protection –

- The OS ensures that all access to system resources is controlled.
- The OS ensures that external I/O devices are protected from invalid access attempts.
- The OS provides authentication features for each user by means of passwords.

3) Define system call. And explain any 4 type with examples.

A)

system call Definition: "system call is a request from an application program for the operating system to perform some hardware action on behalf of the application. System calls are initiated with a software interrupt assembly language instruction".



The system call provides an interface to the operating system services. Application developers often do not have direct access to the system calls, but can access them through an application programming interface (API). The functions that are included in the API invoke the actual system calls. By using the API, certain benefits can be gained:

- Portability: as long a system supports an API, any program using that API can compile and run.
- Ease of Use: using the API can be significantly easier than using the actual system call.

• Types of System Calls

There are 5 different categories of system calls:

1.process control, 2.file manipulation, 3.device manipulation, 4.information maintenance and 5.communication.

1 Process Control

A running program needs to be able to stop execution either normally or abnormally. When execution is stopped abnormally, often a dump of memory is taken and can be examined with a debugger.

Process control

1. end, abort
2. load, execute
3. create process, terminate process
4. get process attributes, set process attributes
5. wait for time
6. wait event, signal event
7. allocate and free memory

2 File Management

Some common system calls are *create*, *delete*, *read*, *write*, *reposition*, or *close*. Also, there is a need to determine the file attributes – *get* and *set* file attribute. Many times the OS provides an API to make these system calls.

- File management system calls

° create file, delete file
o open, close
° read, write, reposition

o get file attributes, set file attributes

Device Management

Process usually require several resources to execute, if these resources are available, they will be granted and control returned to the user process. These resources are also thought of as devices. Some are physical, such as a video card, and others are abstract, such as a file.

User programs *request* the device, and when finished they *release* the device. Similar to files, we can *read*, *write*, and *reposition* the device.

- Device management system calls

o request device, release device
° read, write, reposition
o get device attributes, set device attributes

° logically attach or detach devices

Information Management

Some system calls exist purely for transferring information between the user program and the operating system. An example of this is *time*, or *date*.

The OS also keeps information about all its processes and provides system calls to report this information.

- Information maintenance system calls

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

o set process, file, or device attributes

Communication

There are two models of interprocess communication, the message-passing model and the shared memory model.

- Message-passing uses a common mailbox to pass messages between processes.
- Shared memory use certain system calls to create and gain access to create and gain access to regions of memory owned by other processes. The two processes exchange information by reading and writing in the shared data.
- Communication system calls
- o create, delete communication connection
- ° send, receive messages
- o transfer status information

o attach or detach remote devices

4 OPERATING SYSTEM OPERATIONS:

If there are no processes to execute, OS will sit idle and wait for some event to happen. Interrupts could be hardware interrupts or software interrupts. The OS is designed to handle both. A trap (or an exception) is a software generated interrupt caused either by an error (e.g. divide by zero) or by a specific request from a user program. A separate code segment is written in the OS to handle different types of interrupts.

These codes are known as interrupt handlers/ interrupt service routine. A properly designed OS ensures that an illegal program should not harm the execution of other programs. To ensure this, the OS operates in dual mode.

Dual mode of operation

The OS is design in such a way that it is capable of differentiating between the execution of OS code and user defined code. To achieve this OS need two different modes of operations this is thereby controlled by mode bit added to hardware of computer system as shown in Table 4.

Mode Type	Definition	Mode Bit	Examples
User Mode	User Defined codes are executed	Mode Bit=1	Creation of word document or in general user using any application program
Kernel Mode	OS system codes are executed (also known as supervisor, system, or privileged mode)	Mode Bit=0	Handling interrupts- Transferring control of a process from CPU to I/O on request

Table: User and Kernel Mode of Operating System

Transition from User to Kernel mode

When a user application is executing on the computer system OS is working in user mode. On signal of system call via user application, the OS transits from user mode to kernel mode to service that request as shown below.

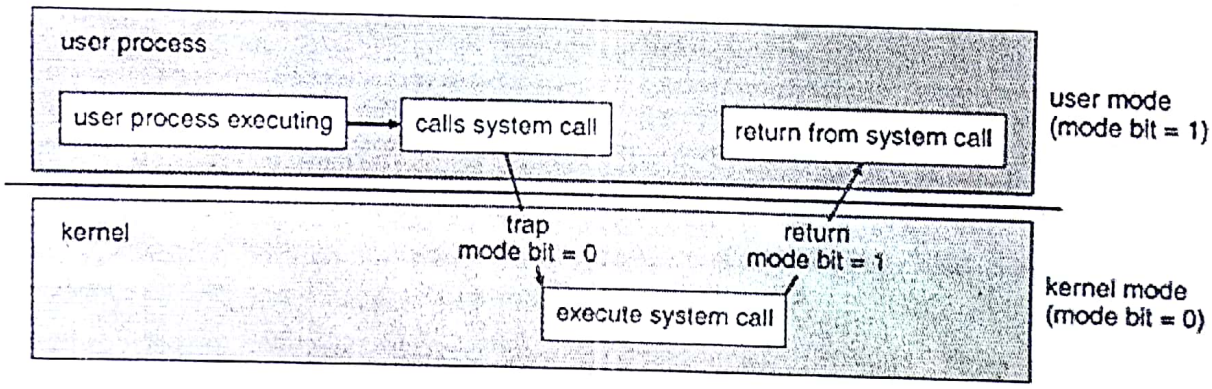


Fig: Transition from user to kernel mode

When the user starts the system the hardware starts in monitor/ kernel mode and loads the operating system. OS has the initial control over the entire system, when instructions are executed in kernel mode. OS then starts the user processes in user mode and on occurrence of trap, interrupt or system call again switch to kernel mode and gains control of the system. System calls are designed for the user programs through which user can ask OS to perform tasks reserved for operating system. System calls usually take the form of the trap. Once the OS service the interrupt it transfers control back to user program hence user mode by setting mode bit=1.

Process Management

A program does nothing unless its instructions are executed by a CPU. A program in execution, as mentioned, is a process. compiler is a process, A word-processing is a process. A process needs certain resources—including CPU time, memory, files, and I/O devices—to accomplish its task. These resources

are either given to the process when it is created or allocated to it while it is running. a program is a *passive* entity, such as the contents of a file stored on disk, whereas a process is an *active* entity. A single-threaded process has one **program counter** specifying the next instruction to execute.
The operating system is responsible for the following activities in connection with 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.**

Memory Management:

Operating System also Manages the Memory of the Computer System means Provide the Memory to the Process and Also Deallocate the Memory from the Process. And also defines that if a Process gets completed then this will deallocate the Memory from the Processes. Main memory is a repository of quickly accessible data shared by the CPU and I/O devices. The central processor reads instructions from main memory during the instruction-fetch cycle and both reads and writes data from main memory during the data-fetch cycle

The operating system is responsible for the following activities in connection with memory management:

- **Keeping track of which parts of memory are currently being used and by whom**
- **Deciding which processes (or parts thereof) and data to move into and out of memory**
- **Allocating and deallocating memory space as needed**

File-System Management:

A file is a collection of related information defined by its creator. Commonly, files represent programs (both source and object forms) and data. Operating System also Controls the all the Storage Operations means how the data or files will be Stored into memory the and how the Files will be Accessed by the users etc. All the Operations those are Responsible for Storing and Accessing the Files is determined by the Operating System Operating System also Allows us Creation of Files, Creation of Directories and Reading and Writing the data of Files and Directories and also Copy the contents of the Files and the Directories from One Place to Another Place.

Creating and deleting files

- Creating and deleting directories to organize files
- Supporting primitives for manipulating files and directories
- Mapping files onto secondary storage
- Backing up files on stable (nonvolatile) storage media

Mass-Storage Management:

Most programs—including compilers, assemblers, word processors, editors, and formatters—are stored on a disk until loaded into memory and then use the disk as both the source and destination of their processing. Hence, the proper management of disk storage is of central importance to a computer system.
The operating system is responsible for the following activities in connection with disk management:

- Free-space management
- Storage allocation
- Disk scheduling

Because secondary storage is used frequently, it must be used efficiently. The entire speed of operation of a computer may hinge on the speeds of the disk subsystem and of the algorithms that manipulate that subsystem.

Caching: it is an important principle of computer systems. As all we know all information is kept in some storage such as **main memory**, as it is used, it is copied into a faster storage system-the **cache**-on a temporary basis. When we need a particular piece of information, we first check whether it is in cache or not, if it is in the cache we use it directly from the cache, otherwise we use the information from the source from where it is present and made a copy of that information in the cache assuming that we will need it again soon. By doing so we decreases the searching time of that particular information.

Protection and Security:

Protection is a mechanism for controlling the access of processes or users to the resources defined by a computer system. Protection can improve reliability by detecting latent errors at the interfaces between component subsystems. Early detection of interface errors can often prevent contamination of a healthy subsystem by another subsystem that is malfunctioning. Protection and security require the system to be able to distinguish among all its users. Most operating systems maintain a list of user names and associated **user identifiers (user IDs).**

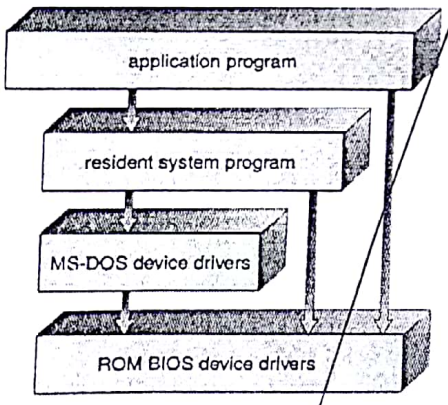
Group functionality can be implemented as a system-wide list of group names and **group identifiers.** A user can be in one or more groups, depending on operating-system design decisions. The user's group IDs are also included in every associated process and thread.

5 **MS-DOS System Structure**

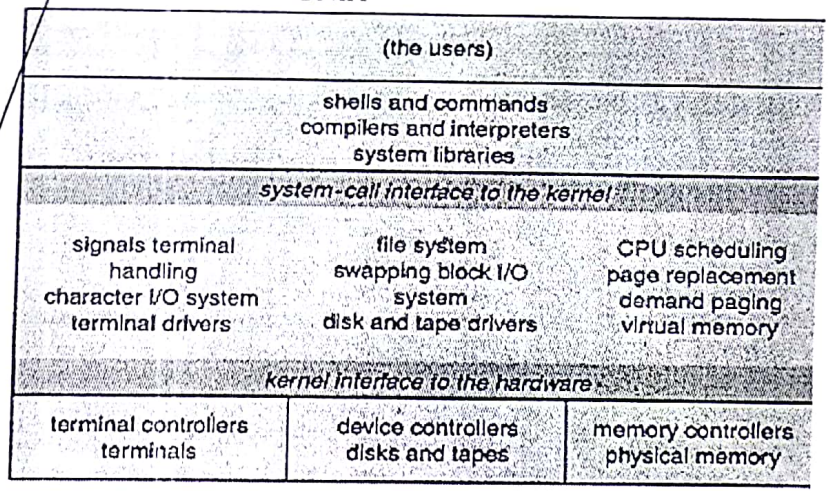
MS-DOS – written to provide the most functionality in the least space
1. not divided into modules
2. Although MS-DOS has some structure, its interfaces and levels of functionality are not well separated

In MS-DOS, the interfaces and levels of functionality are not well separated. For instance, application programs are able to access the basic I/O routines to write directly to the display and disk drives. Such freedom leaves MS-DOS vulnerable to errant (or malicious) programs, causing entire system crashes when user programs fail.

MS-DOS Layer Structure



UNIX



the original UNIX operating system. UNIX is another system that initially was limited by hardware functionality. Everything below the system call interface and above the physical hardware is the kernel. The kernel provides the file system, CPU scheduling, memory management, and other operating-system functions through system calls.

UNIX -- limited by hardware functionality, the original UNIX operating system had limited structuring. The UNIX OS consists of two separable parts.

1. Systems programs
2. The kernel
3. Consists of everything below the system-call interface and above the physical hardware
4. Provides the file system, CPU scheduling, memory management, and other operating-system functions.

5.a large number of functions for one level.

Layered Approach:

A system can be made modular in many ways. One method is the **layered approach,** in which the operating system is broken up into a number of layers (levels). The bottom layer (layer 0) id the hardware; the highest (layer N) is the user interface.

The main advantage of the layered approach is simplicity of construction and debugging. The layers are selected so that each uses functions (operations) and services of only lower-level layers. This approach simplifies debugging and system verification. The first layer can be debugged without any concern for the rest of the system, because, by definition, it uses only the basic hardware (which is assumed correct) to implement its functions. Once the first layer is debugged, its correct functioning can be assumed while the second layer is debugged, and so on. If an error is found during debugging of a particular layer, the error must be on that layer, because the layers below it are already debugged. Thus, the design and implementation of the system is simplified.

Advantages of layered operating systems are:

1. It is decomposable and therefore effects separation of concerns and different abstraction levels
2. It allows good maintenance, where you can make changes without affecting layer interfaces

- Disadvantages of layered operating systems are:

1. It is difficult to exactly assign of functionalities to the correct and appropriate layer.
2. Because of having too many layers, performance of the system is degraded.)5

6c

About kernel:

Kernel is the core part of an operating system; it manages the system resources. Kernel is like a bridge between application and hardware of the computer. The Kernel can be classified further into two categories, Microkernel and Monolithic Kernel.

Definition of Microkernel:

Microkernel being a kernel manages all system resources. But in a microkernel, the **user services** and the **kernel services** are implemented in different address space. The user services are kept in **user address space**, and kernel services are kept under **kernel address space**. This **reduces** the size of the kernel and further reduces the size of operating system.

In addition to the communication between application and hardware of the system, the microkernel provides minimal services of process and memory management. The communication between the client program/application and services running in user address space is established through message passing. They never interact directly. This reduces the speed of execution of microkernel.

In a microkernel, the user services are isolated from kernel services so if any user service fails it does not affect the kernel service and hence Operating system **remain unaffected**. This is one of the advantages in the microkernel. The microkernel is easily **extendable**. If the new services are to be added, they are added to user address space and hence, the kernel space do not require any modification. The microkernel is also easily portable, secure and reliable.

Comparison Chart

BASIS FOR COMPARISON	MICROKERNEL	MONOLITHIC KERNEL
Basic	In microkernel user services and	In monolithic kernel, both kernel, user services and kernel

BASIS FOR COMPARISON	MICROKERNEL	MONOLITHIC KERNEL
	services are kept in separate address space.	services are kept in the same address space.
Size	Microkernel are smaller in size.	Monolithic kernel is larger than microkernel.
Execution	Slow execution.	Fast execution.
Extendible	The microkernel is easily extendible.	The monolithic kernel is hard to extend.
Security	If a service crashes, it does effect on working of microkernel.	If a service crashes, the whole system crashes in monolithic kernel.
Code	To write a microkernel, more code is required.	To write a monolithic kernel, less code is required.
Example	QNX, L4Linux,Mac os x, Minix.	Linux, Windows (95,98), Solaris.

Multi programming: Multiprogramming is the technique of running several programs at a time using timesharing. It allows a computer to do several things at the same time. Multiprogramming creates logical parallelism. The concept of multiprogramming is that the operating system keeps several jobs in memory simultaneously.

Multitasking

Multitasking has the same meaning of multiprogramming but in a more general sense, as it refers to having multiple (programs, processes, tasks, threads) running at the same time. This term is used in modern operating systems when multiple tasks share a common processing resource (e.g., CPU and Memory). At any time the CPU is executing one task only while other tasks waiting their turn. The

illusion of parallelism is achieved when the CPU is reassigned to another task (i.e. *process* or *thread context switching*).

There are subtle differences between multitasking and multiprogramming. A *task* in a multitasking operating system is not a whole application program but it can also refer to a "thread of execution" when one process is divided into sub-tasks. Each smaller task does not hijack the CPU until it finishes like in the older multiprogramming but rather a fair share amount of the CPU time called quantum.

MODELS OF DISTRIBUTED SYSTEMS IN OS

Client-Server

The client-server model is probably the most popular paradigm. The server is responsible for accepting, processing, and replying to requests. It is the producer. The client is purely the consumer. It requests the services of the server and accepts the results.

The basic web follows the client-server model. Your browser is the client. It requests web pages from a server (e.g., google.com), waits for results, and displays them for the user.

Peer-To-Peer Model

The peer-to-peer model assumes that each entity in the network has equivalent functionality. In essence, it can play the role of a client or a server. Ideally, this reduces bottlenecks and enables each entity to contribute resources to the system. Unfortunately, it doesn't always work that way. enabling communication in such a system is challenging. First, peers must locate other peers in order to participate in the system. This is rarely done in a truly distributed or peer-to-peer fashion.

Network Computers/Thin Clients

The network computer model assumes that the end user machine is a low-end computer that maintains a minimal OS. When it boots, it retrieves the OS and files/applications from a central server and runs applications locally. The thin client model is similar, though assumes that the process runs remotely and the client machine simply displays results (e.g., X-windows and VNC).

This model has been around for quite some time, but has recently received much attention. Google and Amazon are both promoting "cloud computing". Sun's Sun Ray technology also makes for an interesting demonstration. Though this model has yet to see success, it is beginning to look more promising.

Mobile Devices

There is an increasing need to develop distributed systems that can run atop devices such as cell phones, cameras, and MP3 players. Unlike traditional distributed computing entities, which communicate over the Internet or standard local area networks, these devices often communicate via wireless technologies such as Bluetooth or other low bandwidth and/or short range mechanisms. As a result, the geographic location of the devices impacts system design. Moreover, mobile systems must take care to consider the battery constraints of the participating devices. System design for mobile ad hoc networks (MANETs), sensor networks, and delay/disruption tolerant networks (DTNs) is a very active area of research.