UNIT-1

An operating System (OS) is an intermediary between users and computer hardware. It provides an environment in which a user can execute programs conveniently and efficiently. In technical terms, it is system software which manages hardware. An operating System controls the allocation of resources and services such as memory, processors, devices and information.

Operating System objectives:

An amazing aspect of operating systems is how varied they are in accomplishing these tasks. Mainframe operating systems are designed primarily to optimize utilization of hardware. Personal computer (PC) operating systems support complex games, business applications, and everything in between. Operating systems for handheld computers are designed to provide an environment in which a user can easily interface with the computer to execute programs. Thus, some operating systems are designed to be convenient, others to be efficient, and others some combination of the two.

- Execute user programs and make solving user problems easier
- Make the computer system convenient to use
- Use the computer hardware in an efficient manner

User View:

The user's view of the computer varies according to the interface being used. 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. In this case/ the operating system is designed mostly for ease of use with some attention paid to performance and none paid to resource utilization various hardware and software resources are shared. Performance is, of course, important to the user; but such systems are optimized for the single-user experience rather than the requirements of multiple users. 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. The operating system in such casesis designed to maximize resource utilization to assure that all available CPU time, memory, and I/0 are used efficiently and that no individual user takes more than her fair share.

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. Therefore, their operating system is designed to compromise

between individual usability and resource utilization. Recently, many varieties of handheld computers have come into fashion. Most of these devices are standalone units for individual users. 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. Their operating systems are designed mostly for individual usability, but performance per unit of battery life is important as well. Some computers have little or no user view. For example, embedded computers in home devices and automobiles may have numeric keypads and may turn indicator lights on or off to showstatus, but they and their operating systems are designed primarily to run without user intervention.

System View:

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/0 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. As we have seen, resource allocation is especially important where many users access the same mainframe or minicomputer. A slightly different view of an operating system emphasizes the need to control the various I/0 devices and user programs. 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.

Definition

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

OS Functions:

Following are some of the important functions of an operating System.

- Memory Management
- Processor Management
- Device Management

- File Management
- Security
- Control over system performance
- Job accounting
- Error detecting and Prevention
- Coordination between other software and users

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 access directly by the CPU. So for a program to be executed, it must be in the main memory.

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 part are not in use.
- In multiprogramming, OS decides which process will get memory when and how much.
- Allocates the memory when the process requests it to do so.
- De-allocates the memory when the process no longer needs it or has been terminated.

Processor Management

In a multiprogramming environment, OS decides which process gets the processor when and how much time. This function is called process scheduling. The Operating System does the following activities for processor management.

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

Device Management

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

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

File Management

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions. 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.
- Decides who gets the resources.
- Allocates the resources.
- De-allocates the resources.

Following are some of the important activities that Operating System does:

- Security -- By means of password and similar other techniques, preventing unauthorized access to programs and data.
- Control over system performance -- Recording delays between request for a service and response from the system.
- Job accounting -- Keeping track of time and resources used by various jobs and users.
- Error detecting aids -- Production of dumps, traces, error messages and other debugging and error detecting aids.
- Coordination between other Softwares and users -- Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

Computer System Architecture:

Computer system can be divided into four components

· Hardware – provides basic computing resources

CPU, memory, I/O devices

· Operating system

Controls and coordinates use of hardware among various applications and users

· Application programs – define the ways in which the system resources are used to solve the computing problems of the users

Word processors, compilers, web browsers, database systems, video games

· Users

People, machines, other computers

Operating-System Operations

- Dual-mode operation allows OS to protect itself and other system components
- User mode and kernel mode
- Mode bit provided by hardware

1. Provides ability to distinguish when the system is running user code or kernel code.
2. Some instructions designated as privileged, only executable in kernel mode.
3. System call changes mode to kernel, return from call resets it to user.

Evaluation of Operating Systems

1. Simple Batch System

2. Multiprogramming Batch System
3. Multiprocessor System Or parallel system
4. Desktop System
5. Distributed Operating System
6. Clustered System
7. Realtime Operating System
8. Handheld System

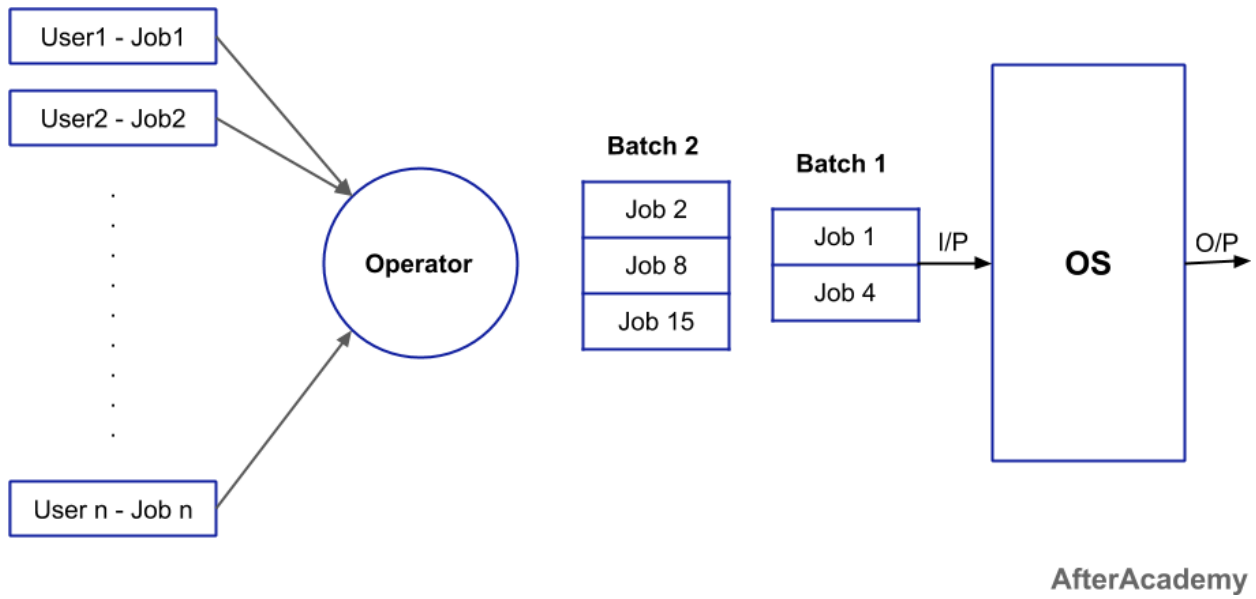
Simple Batch Systems

In this type of system, there is no direct interaction between user and the computer.
The user has to submit a job to a computer operator.
Then the computer operator places a batch of several jobs on an input device.
Jobs are batched together by type of languages and requirements.
Then a special program, the monitor, manages the execution of each program in the batch.
The monitor is always in the main memory and available for execution.

Advantages of Simple Batch Systems

- No interaction between user and computer.
- No mechanism to prioritise the processes.

Simple Batch Systems



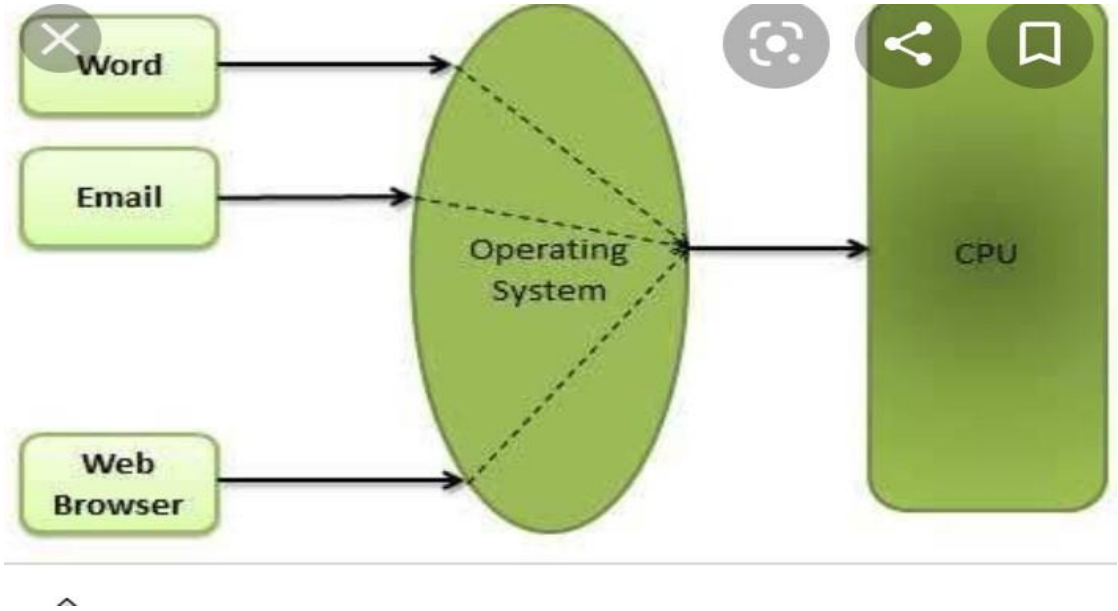
Multiprogramming Batch Systems

In this the operating system picks up and begins to execute one of the jobs from memory.
Once this job needs an I/O operation, the operating system switches to another job (CPU and OS are always busy).
Jobs in the memory are always less than the number of jobs on disk(Job Pool).

If several jobs are ready to run at the same time, then the system chooses which one to run through the process of CPU Scheduling.
In a Non-multiprogrammed system, there are moments when the CPU sits idle and does not do any work.
In a Multiprogramming system, the CPU will never be idle and keeps on processing.
Time Sharing Systems are very similar to Multiprogramming batch systems. In fact time sharing systems are an extension of multiprogramming systems.

In Time sharing systems the prime focus is on minimizing the response time, while in multiprogramming the prime focus is to maximize the CPU usage.

Multiprogramming Batch Systems

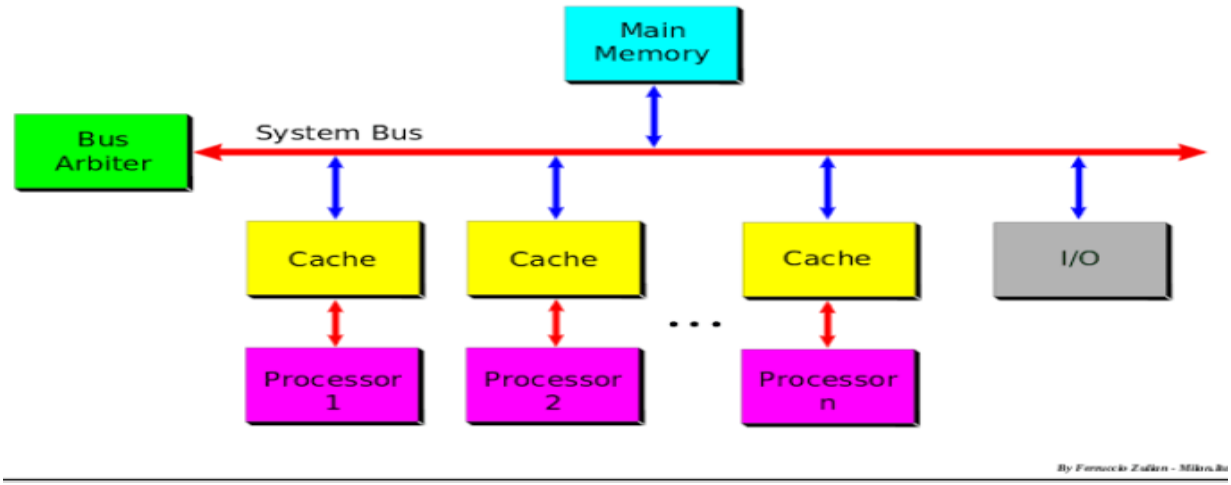


Multiprocessor Systems or parallel system
A Multiprocessor system consists of several processors that share a common physical memory. Multiprocessor system provides higher computing power and speed. In multiprocessor system all processors operate under single operating system. Multiplicity of the processors and how they do act together are transparent to the others.

Advantages of Multiprocessor Systems

- Enhanced performance
- Execution of several tasks by different processors concurrently, increases the system's throughput without speeding up the execution of a single task.If possible, system divides task into many subtasks and then these subtasks can be executed in parallel in different processors. Thereby

speeding up the execution of single tasks.



Desktop Systems or personal computer

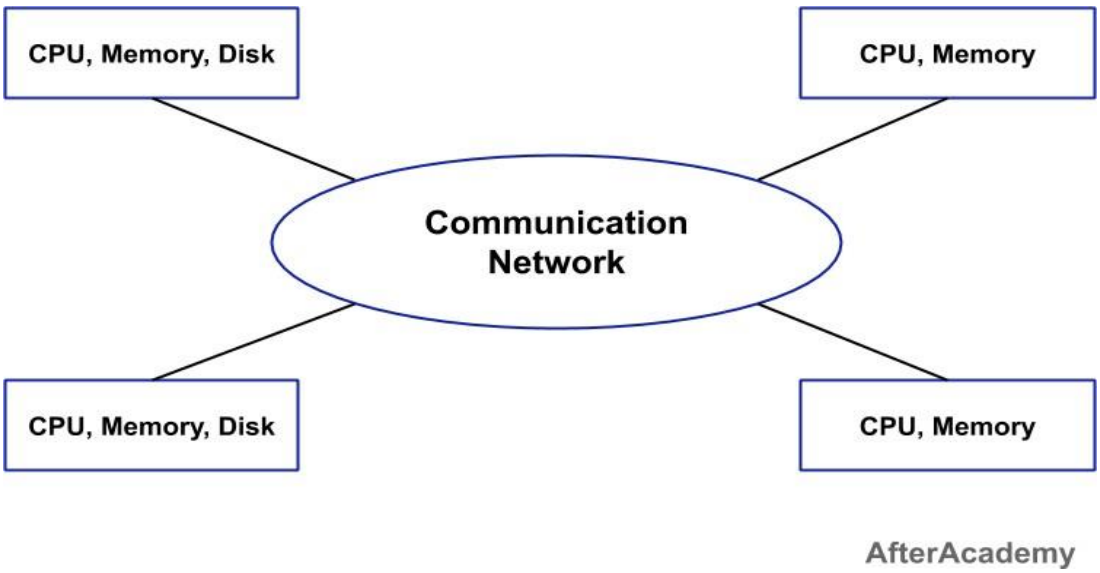
Earlier, CPUs and PCs lacked the features needed to protect an operating system from user programs. PC operating systems therefore were neither multiuser nor multitasking. However, the goals of these operating systems have changed with time; instead of maximizing CPU and peripheral utilization, the systems opt for maximizing user convenience and responsiveness. These systems are called Desktop Systems and include PCs running Microsoft Windows and the Apple Macintosh. Operating systems for these computers have benefited in several ways from the development of operating systems for mainframes.

Microcomputers were immediately able to adopt some of the technology developed for larger operating systems. On the other hand, the hardware costs for microcomputers are sufficiently low that individuals have sole use of the computer, and CPU utilization is no longer a prime concern. Thus, some of the design decisions made in operating systems for mainframes may not be appropriate for smaller systems.

Distributed Operating System

The motivation behind developing distributed operating systems is the availability of powerful and inexpensive microprocessors and advances in communication technology.

These advancements in technology have made it possible to design and develop distributed systems consisting of many computers that are interconnected by communication networks. The main benefit of distributed systems is its low price/performance ratio.



Advantages Distributed Operating System

- As there are multiple systems involved, users at one site can utilize the resources of systems at other sites for resource-intensive tasks.
- Fast processing.
- Less load on the Host Machine.

Types of Distributed Operating Systems

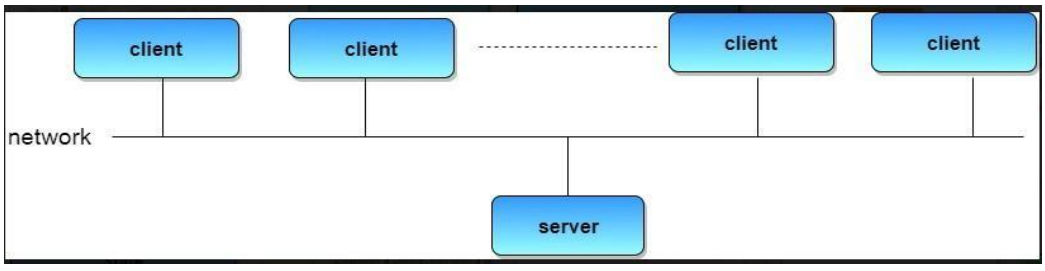
Following are the two types of distributed operating systems used:

Client-Server Systems
Peer-to-Peer Systems

Client-Server Systems

Centralized systems today act as server systems to satisfy requests generated by client systems. The general structure of a client-server system is depicted in the figure below:

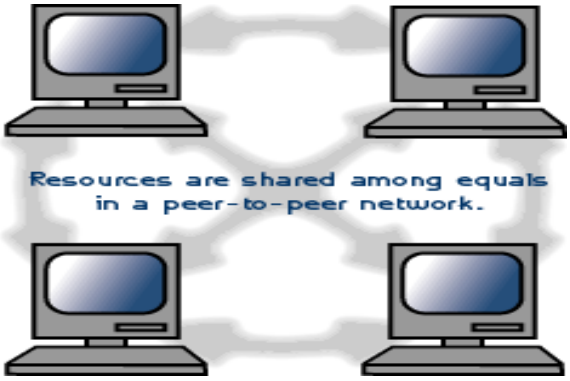
Client-Server Systems



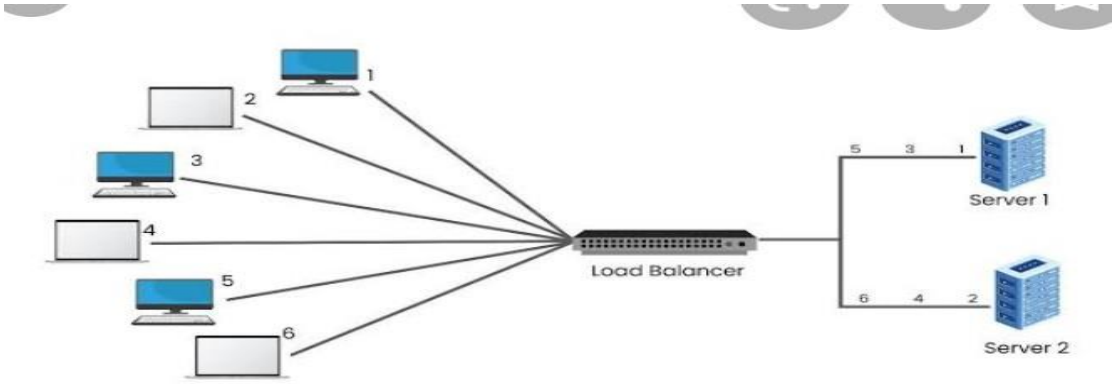
Server Systems can be broadly categorized as: Compute Servers and File Servers.

Compute Server systems provide an interface to which clients can send requests to perform an action, in response to which they execute the action and send back results to the client.
File Server systems, provide a file-system interface where clients can create, update, read, and delete files.

Peer-to-Peer Systems
Peer-to-peer network operating systems allow users to share resources and files located on their computers and to access shared resources found on other computers. However, they do not have a file server or a centralized management source (See fig. 1). In a peer-to-peer network, all computers are considered equal; they all have the same abilities to use the resources available on the network



Clustered Systems
Like parallel systems, clustered systems gather together multiple CPUs to accomplish computational work.



Clustered systems differ from parallel systems, however, in that they are composed of two or more individual systems coupled together.
The definition of the term clustered is not concrete; the general accepted definition is that clustered computers share storage and are closely linked via LAN networking.
Clustering is usually performed to provide high availability.

A layer of cluster software runs on the cluster nodes. Each node can monitor one or more of the others. If the monitored machine fails, the monitoring machine can take ownership of its storage, and restart the application(s) that were running on the failed machine. The failed machine can remain down, but the users and clients of the application would only see a brief interruption of service.
Asymmetric Clustering - In this, one machine is in hot standby mode while the other is running the applications. The hot standby host (machine) does nothing but monitor the active server. If that server fails, the hot standby host becomes the active server.
Symmetric Clustering - In this, two or more hosts are running applications, and they are monitoring each other. This mode is obviously more efficient, as it uses all of the available hardware.
Parallel Clustering - Parallel clusters allow multiple hosts to access the same data on the shared storage. Because most operating systems lack support for this simultaneous data access by multiple hosts, parallel clusters are usually accomplished by special versions of software and special releases of applications.
Clustered technology is rapidly changing. Clustered system's usage and its features should expand greatly as Storage Area Networks(SANs). SANs allow easy attachment of multiple hosts to multiple storage units. Current clusters are usually limited to two or four hosts due to the complexity of connecting the hosts to shared storage.

Real Time Operating System
It is defined as an operating system known to give maximum time for each of the critical operations that it performs, like OS calls and interrupt handling.

The Real-Time Operating system which guarantees the maximum time for critical operations and completes them on time are referred to as Hard Real-Time Operating Systems.

While the real-time operating systems that can only guarantee a maximum of the time, i.e. the critical task will get priority over other tasks, but no assurity of completeing it in a defined time. These systems are referred to as Soft Real-Time Operating Systems.

Handheld Systems
Handheld systems include Personal Digital Assistants(PDAs), Cellular Telephones with connectivity to a network such as the Internet. They are usually of limited size due to which most handheld devices have a small amount of memory, include slow processors, and feature small display screens.

Many handheld devices have between 512 KB and 8 MB of memory. As a result, the operating system and applications must manage memory efficiently. This includes returning all allocated memory back to the memory manager once the memory is no longer being used.
Advantages

- Notes can be digitized
- Used for sending and receiving invoices
- Used for asset management
- For scanning barcodes
- Watch television through the medium of internet by IPTV
- Can sync information or data and share stuff
- Can provide educational materials to others

- It is best suited in terms to meet the mobility needs and capture data. It's completely wireless and it is suited for all types of environments and applications.

There are various features of handheld computers such as:

- They are extremely easy to hold and portable.
- Good connectivity — has the ability to stay connected with various other devices. The connectivity with other network connections is also very strong.
- Individuality- in order to suit the needs of an individual it is very adaptable.
- Socially interactive- in the same environment it maintains a good amount of connectivity in order to collaborate with the different other users.

Differences between Multiprocessing and Multiprogramming

Sr. No.	Multiprocessing Or parallel operating system	Multiprogramming
1	Multiprocessing refers to processing of multiple processes at same time by multiple CPUs.	Multiprogramming keeps several programs in main memory at the same time and executes them concurrently utilizing a single CPU.
2	It utilizes multiple CPUs.	It utilizes a single CPU.
3	It permits parallel processing.	Context switching takes place.
4	Less time taken to process the jobs.	More Time taken to process the jobs.
5	It facilitates much efficient utilization of devices of the computer system.	Less efficient than multiprocessing.
6	Usually more expensive.	Such systems are less expensive.

Differences between Real Time System and Timesharing System

Sr. No.	Real Time System	Timesharing System
1	In this system, events mostly external to the computer system are accepted and processed within certain deadlines.	In this system, many users are allowed to simultaneously share the computer resources.
2	Real time processing is mainly devoted to one application.	Time sharing processing deals with many different applications.
3	Users can make enquiry only and cannot write or modify programs.	Users can write and modify programs.
4	Users must get a response within the specified time limit; otherwise it may result in a disaster.	Users should get a response within fractions of seconds but if not, the results are not disastrous.
5	No context switching takes place in this system.	The CPU switches from one process to another as a time slice expires or a process terminates.

EXAMPLES OF OS

Windows is a series of operating systems developed by Microsoft. Each version of Windows includes a graphical user interface, with a desktop that allows users to view files and folders in windows. For the past two decades, Windows has been the most widely used operating system for personal computers PCs.

Microsoft Windows is designed for both home computing and professional purposes. Past versions of Windows home editions include Windows 3.0 (1990), Windows 3.1 (1992), Windows 95 (1995), Windows 98 (1998), Windows Me (2000), Windows XP

(2001), and Windows Vista (2006).Windows 7, was released in 2009.The current version 20H2, Windows 10, was released in 2020

Best Features of Windows Operating System

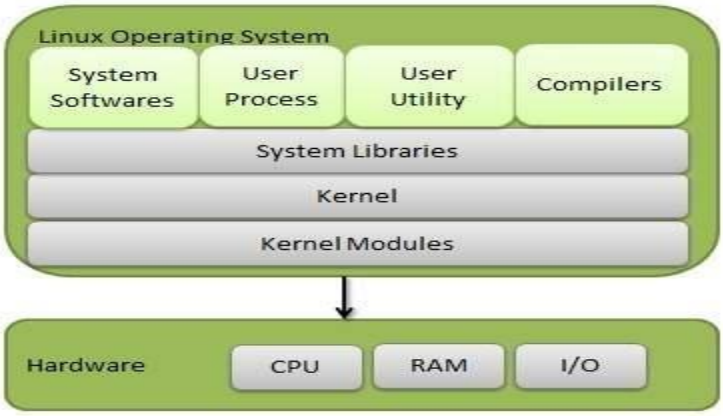
1. Speed. ...
2. Compatibility. ...
3. Lower Hardware Requirements. ...
4. Search and Organization. ...
5. Safety and Security. ...
6. Interface and Desktop. ...
7. Taskbar/Start menu

Linux is one of the popular versions of the UNIX operating System. It is open source as its source code is freely available. It is free to use. Linux was designed considering UNIX compatibility. Its functionality list is quite similar to that of UNIX.

Components of Linux System

Linux Operating System has primarily three components

- Kernel − Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.
- System Library − System libraries are special functions or programs using which application programs or system utilities access Kernel's features. These libraries implement most of the functionalities of the operating system and do not require kernel module's code access rights.
- System Utility − System Utility programs are responsible to do specialized, individual level tasks.



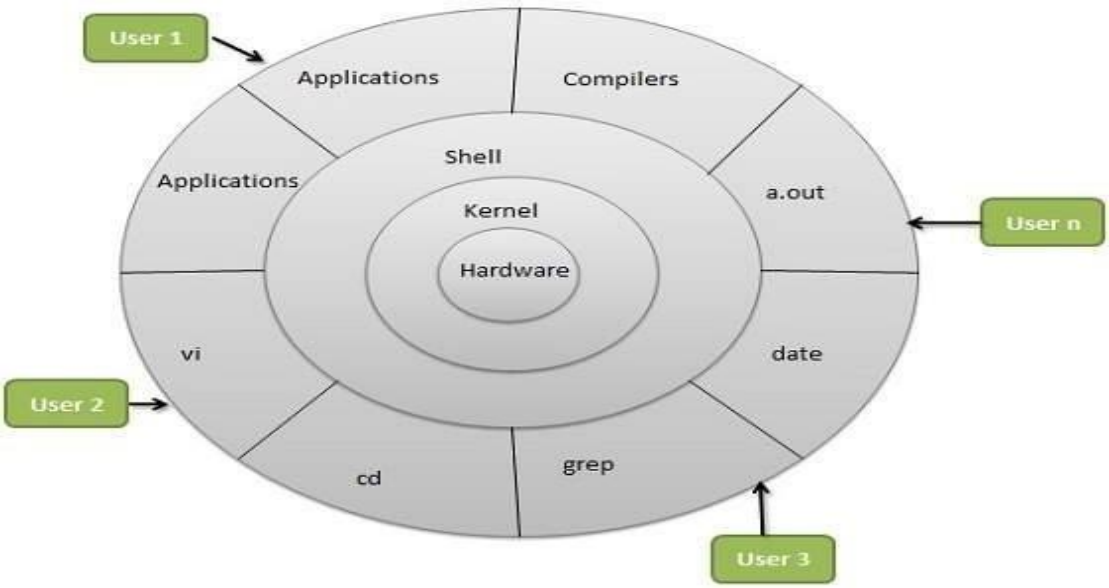
Basic Features

Following are some of the important features of Linux Operating System.

- Portable − Portability means software can work on different types of hardware in the same way. Linux kernel and application programs support their installation on any kind of hardware platform.
- Open Source − Linux source code is freely available and it is a community based development project. Multiple teams work in collaboration to enhance the capability of the Linux operating system and it is continuously evolving.
- Multi-User − Linux is a multiuser system meaning multiple users can access system resources like memory/ ram/ application programs at same time.
- Multiprogramming − Linux is a multiprogramming system meaning multiple applications can run at same time.
- Hierarchical File System − Linux provides a standard file structure in which system files/ user files are arranged.
- Shell − Linux provides a special interpreter program which can be used to execute commands of the operating system. It can be used to do various types of operations, call application programs. etc.
- Security − Linux provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data.

Architecture

The following illustration shows the architecture of a Linux system −



The architecture of a Linux System consists of the following layers −

- Hardware layer − Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).
- Kernel − It is the core component of the Operating System, interacts directly with hardware, provides low level services to upper layer components.
- Shell − An interface to kernel, hiding complexity of kernel's functions from users. The shell takes commands from the user and executes the kernel's functions.
- Utilities − Utility programs that provide the user most of the functionalities of an operating system.

Sr. No.	Key	Linux	Windows
1	Open Source	Linux is Open Source and is free to use.	Windows is not open source and is not free to use.
2	Case sensitivity	The Linux file system is case sensitive.	Windows file system is case insensitive.
3	kernel type	Linux uses a monolithic kernel.	Windows uses a micro kernel.
4	Efficiency	Linux is more efficient in operations as compared to Windows.	Windows is less efficient in operations.

5	Path Separator	Linux uses forward slash as path separator between directories.	Windows uses backward slash as a path separator.
6	Security	Linux is highly secure as compared to Windows.	Windows provides less security as compared to Linux.

OPERATING SYSTEM SERVICES

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

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.

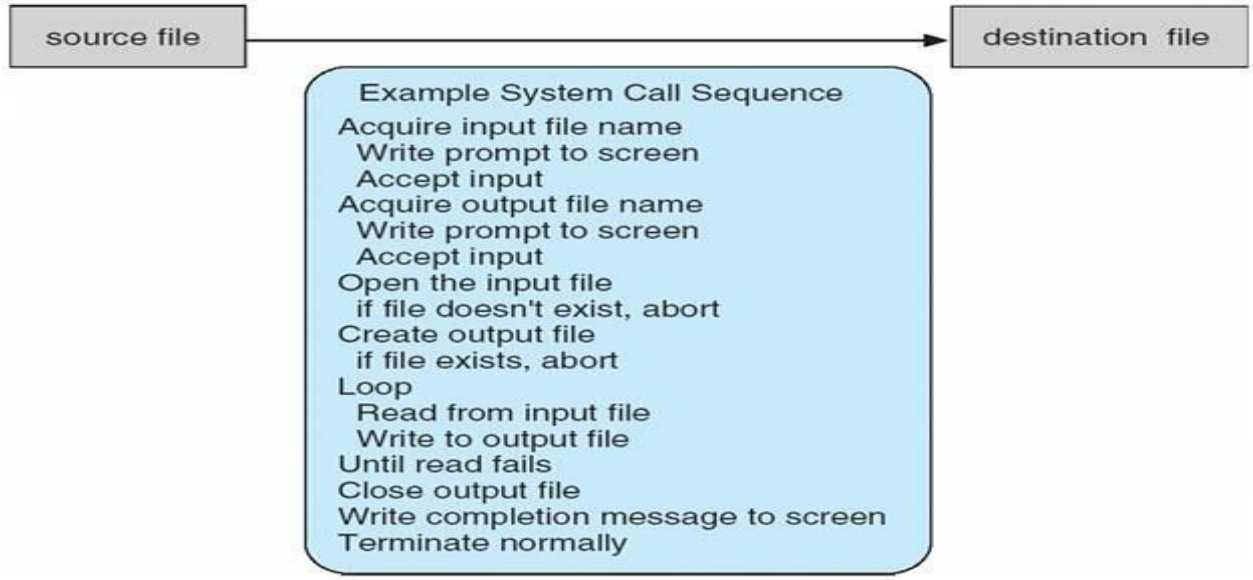
Protection refers to a mechanism or a way to control the access of programs, processes, or users to the resources defined by a computer system. Following are the major activities of an operating system with respect to protection −

- 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.

System Calls

- Programming interface to the services provided by the OS
- Typically written in a high-level language (C or C++)
- Mostly accessed by programs via a high-level Application Program Interface (API) rather than direct system call use
- Three most common APIs are Win32 API for Windows, POSIX API for POSIX-based systems (including virtually all versions of UNIX, Linux, and Mac OS X), and Java API for the Java virtual machine (JVM)

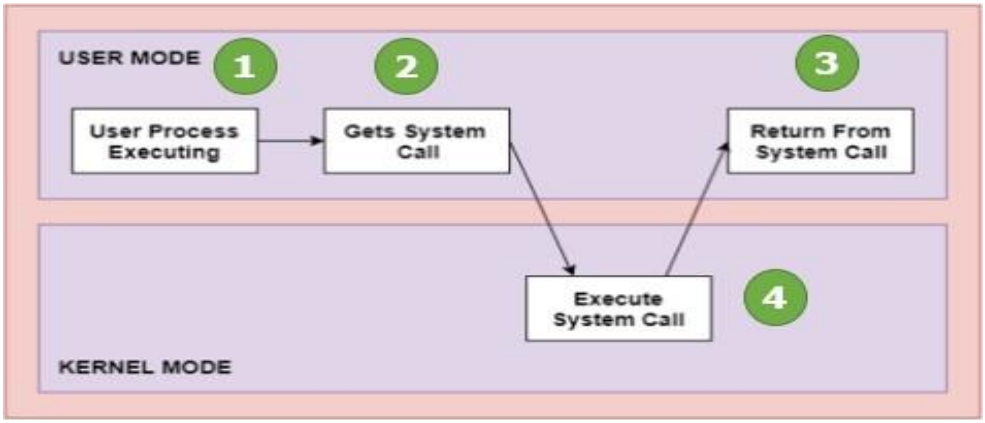
Example of System Calls



System Call Implementation

- Typically, a number associated with each system call
- System-call interface maintains a table indexed according to these Numbers
- The system call interface invokes intended system call in OS kernel and returns status of the system call and any return values
- The caller need know nothing about how the system call is implemented
- Just needs to obey API and understand what OS will do as a result call
- Most details of OS interface hidden from programmer by API

Managed by run-time support library (set of functions built into libraries included with compiler)



Types of System Calls

- Process control
- File management
- Device management
- Information maintenance
- Communications
- Protection

Examples of Windows and Unix System Calls

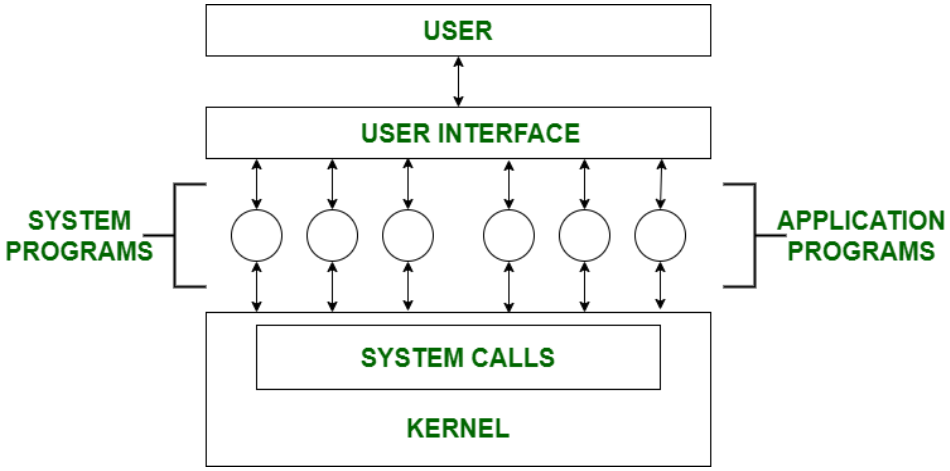
Types of System Calls	Windows	Linux
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Management	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()

Device Management	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap( )

System Programs

System programs provide a convenient environment for program development and execution. These can be divided into:

- File manipulation
- Status information
- File modification
- Programming language support
- Program loading and execution
- Communications
- Application programs



1. **File Management –**

   A file is a collection of specific information stored in the memory of a computer system. File management is defined as the process of manipulating files in the computer system, its management includes the process of creating, modifying and deleting files.

   - It helps to create new files in the computer system and place them at specific locations.
   - It helps in easily and quickly locating these files in the computer system.
   - It makes the process of sharing files among different users very easy and user-friendly.
   - It helps to store files in separate folders known as directories.
   - These directories help users to search files quickly or to manage files according to their types of uses.
   - It helps users to modify the data of files or to modify the name of files in directories.

2. **Status Information –**

   Information like date, time amount of available memory, or disk space is asked by some users. Others provide detailed performance, logging, and debugging information which is more complex. All this information is formatted and displayed

on output devices or printed. Terminal or other output devices or files or a window of GUI is used for showing the output of programs.

3. **File Modification –**

   For modifying the contents of files we use this. For Files stored on disks or other storage devices, we used different types of editors. For searching contents of files or performing transformations of files we use special commands.

4. **Programming-Language support –**

   For common programming languages, we use Compilers, Assemblers, Debuggers, and interpreters which are already provided to users. It provides all support to users. We can run any programming language. All languages of importance are already provided.

5. **Program Loading and Execution –**

   When the program is ready after Assembling and compilation, it must be loaded into memory for execution. A loader is part of an operating system that is responsible for loading programs and libraries. It is one of the essential stages for starting a program. Loaders, relocatable loaders, linkage editors, and Overlay loaders are provided by the system.

6. **Communications –**

   Virtual connections among processes, users, and computer systems are provided by programs. Users can send messages to another user on their screen, users can send email, browse on web pages, remote login, the transformation of files from one user to another.

Some examples of system programs in O.S. are –

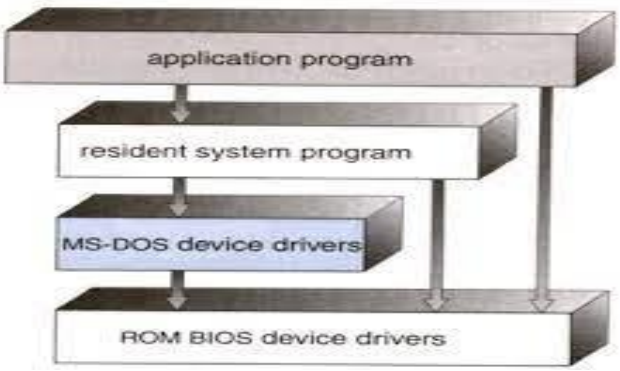
- Anti-virus
- Disk formatting
- Computer language translators

Operating System Structure:

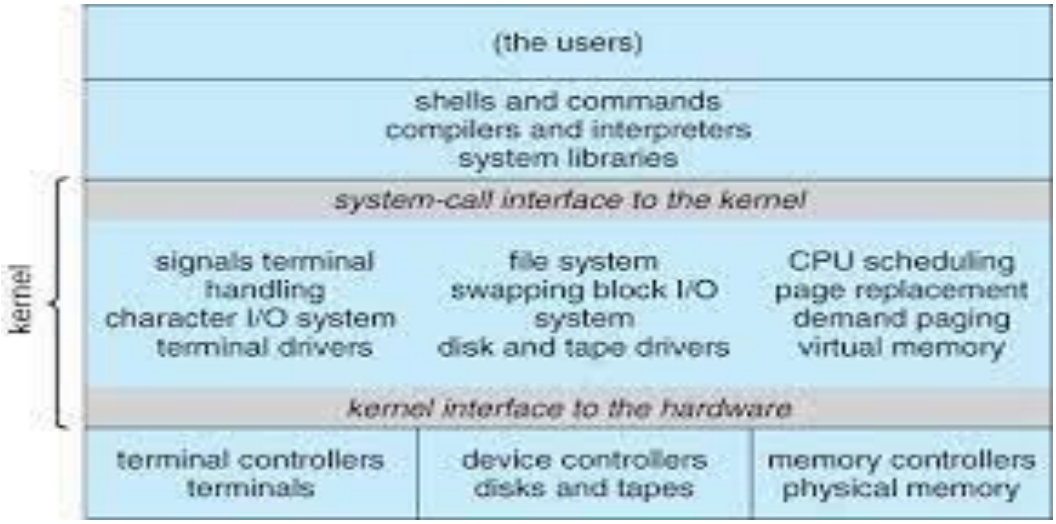
1. Simple Structure

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

MS-DOS Layer Structure



Traditional UNIX System Structure



UNIX

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

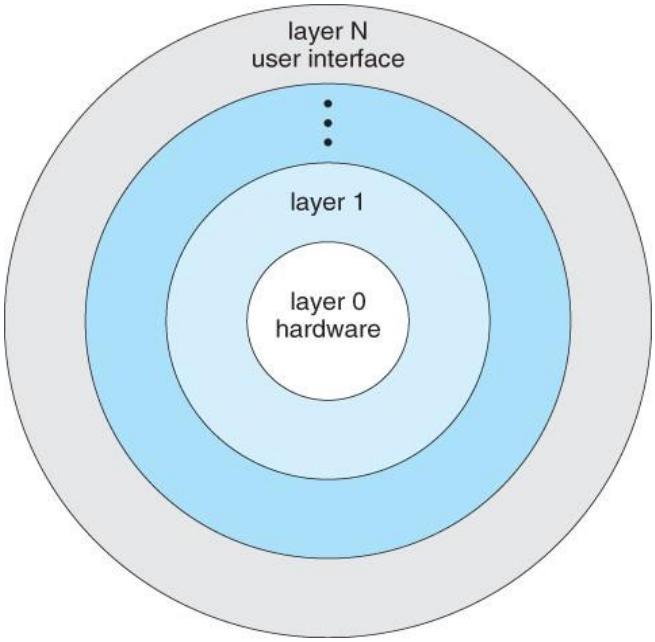
Consists of everything below the system-call interface and above the physical hardware

Provides the file system, CPU scheduling, memory management, and other operating-system functions; a large number of functions for one level

2. Layered Approach

- The operating system is divided into a number of layers (levels), each built on top of lower layers. The bottom layer (layer 0), is the hardware; the highest (layer N) is the user interface.
- With modularity, layers are selected such that each uses functions (operations) and services of only lower-level layers

Layered Operating System



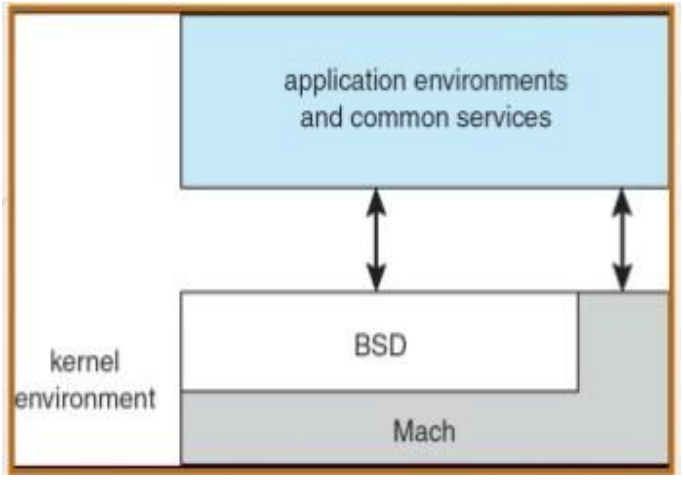
3. Microkernel System Structure

- Moves as much from the kernel into "*user*" space
- Communication takes place between user modules using message passing

Benefits:

- Easier to extend a microkernel
- Easier to port the operating system to new architectures
- More reliable (less code is running in kernel mode)
- More secure
- Performance overhead of user space to kernel space communication

Mac OS X Structure



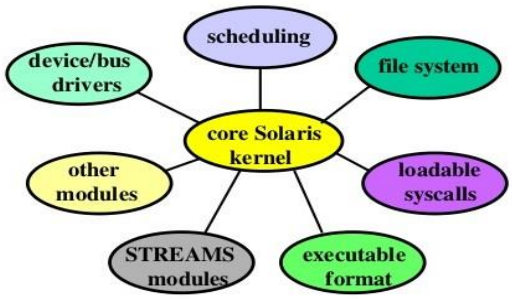
4. Module Structure

- Most modern operating systems implement kernel modules
- Uses object-oriented approach
- Each core component is separate

- Each talks to the others over known interfaces
- Each is loadable as needed within the kernel
- Overall, similar to layers but with more flexible

Solaris Modular Approach

- **The OO technology can be used to create a modular kernel.**
- **The kernel has a set of core component and dynamically links in additional services either during boot time or during run time.**



UNIT-2

Process: A process is a program in execution. The execution of a process must progress in a sequential fashion. Definition of process is following. A process is defined as an entity which represents the basic unit of work to be implemented in the system. Components of process are following. A program is passive entity whereas a process is an active entity. A program can be transformed to process in 2 ways: by double clicking the file and typing the file from command prompt.
Process memory is divided into four sections as shown:

- The text section comprises the compiled program code, read in from non-volatile storage when the program is launched.
- The data section stores global and static variables, allocated and initialized prior to executing main.
- The heap is used for dynamic memory allocation, and is managed via calls to new, delete, malloc, free, etc.
- The stack is used for local variables. Space on the stack is reserved for local variables when they are declared (at function entrance or elsewhere, depending on the language), and the space is freed up when the variables go out of scope. Note that the stack is also used for function return values, and the exact mechanisms of stack management may be language specific.
- Note that the stack and the heap start at opposite ends of the process's free space and grow towards each other. If they should ever meet, then either a stack overflow error will occur, or else a call to new or malloc will fail due to insufficient memory available.
- When processes are swapped out of memory and later restored, additional information must also be stored and restored. Key among them are the program counter and the value of all program registers.