UNIT-1

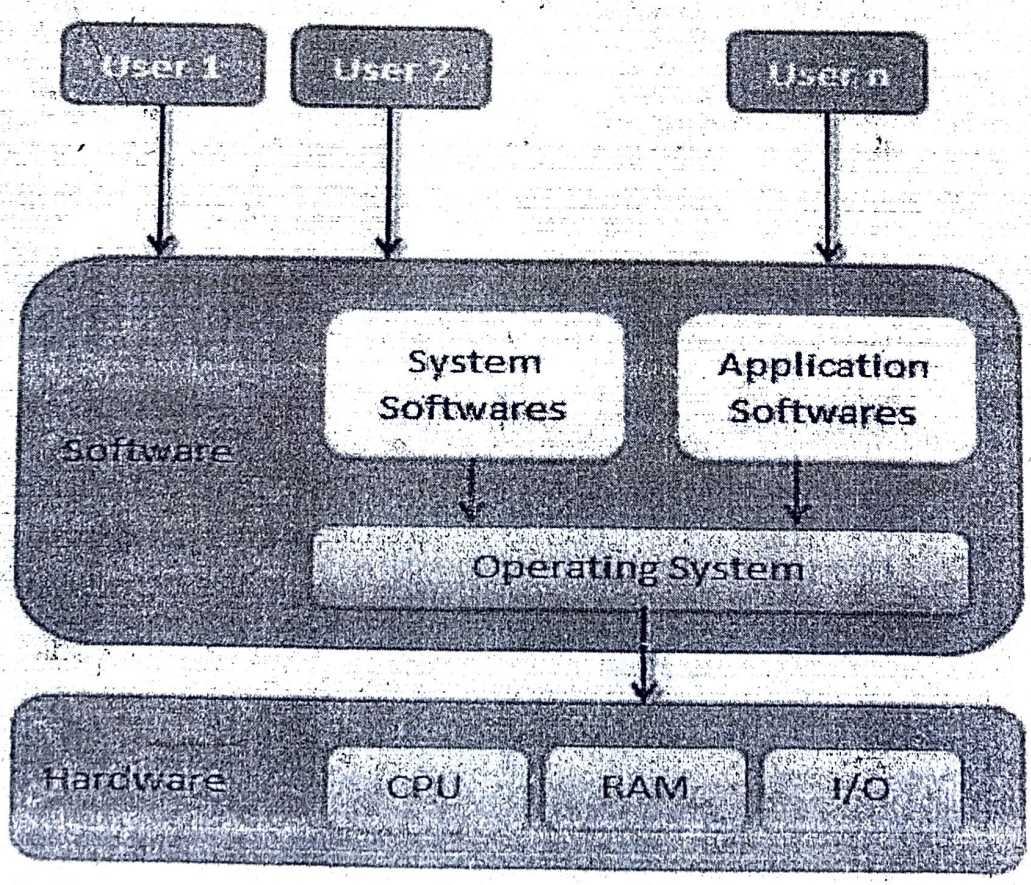
Operating Systems Overview: Operating system functions, Operating system structure, Operating systems operations, Computing environments, Open-Source Operating Systems. System Structures: Operating System Services, User and Operating-System Interface, systems calls, Types of System Calls, system programs, operating system structure, operating system debugging, System Boot.

TOPIC1: What is An Operating System (OS) : is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc.

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.



1.1 Goals of the Operating System

There are two types of goals of an Operating System i.e. Primary Goals and Secondary Goal.

- **Primary Goal:** The primary goal of an Operating System is to provide a user-friendly and convenient environment. We know that it is not compulsory to use the Operating System, but things become harder when the user has to perform all the process scheduling and converting the user code into machine code is also very difficult. So, we make the use of an Operating System to act as an intermediate between us and the hardware. All you need to do is give commands to the Operating System and the Operating System will do the rest for you. So, the Operating System should be convenient to use.
- **Secondary Goal:** The secondary goal of an Operating System is efficiency. The Operating System should perform all the management of resources in such a way that the resources are fully utilised and no resource should be held idle if some request to that resource is there at that instant of time.

So, in order to achieve the above primary and secondary goals, the Operating System performs a number of functions.

1.2 Need of Operating System:

- **OS as a platform for Application programs:**
  Operating system provides a platform, on top of which, other programs, called application programs can run. These application programs help the users to perform a specific task easily. It acts as an interface between the computer and the user. It is designed in such a manner that it operates, controls, and executes various applications on the computer.

- **Managing Input-Output unit:**
  Operating System also allows the computer to manage its own resources such as memory, monitor, keyboard, printer, etc. Management of these resources is required for effective utilization. The operating system controls the various system input-output resources and allocates them to the users or programs as per their requirement.

- **Consistent user interface:**
  Operating System provides the user an easy-to-work user interface, so the user doesn't have to learn a different UI every time and can focus on the content and be productive as quickly as possible. Operating System provides templates, UI components to make the working of a computer, really easy for the user.

- **Multitasking:**
  Operating System manages memory and allows multiple programs to run in their own

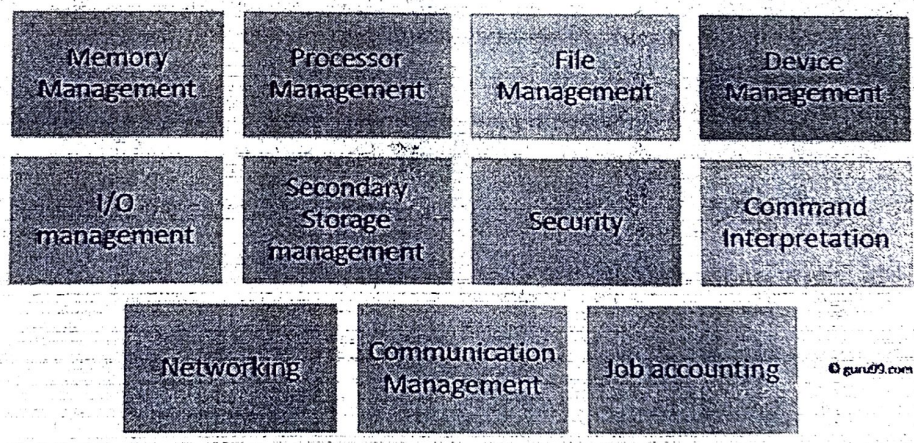
space and even communicate with each other through shared memory. Multitasking gives users a good experience as they can perform several tasks on a computer at a time.

TOPIC: 2Operating system functions:

Following are some of important functions of an operating System.

- Memory Management
- Processor Management
- Device Management
- File Management
- Security
- Control over system performance
- Job accounting
- Error detecting aids
- Coordination between other software and users

Below are the main functions of Operating System:



Functions of Operating System

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 part are not in use.

- In multiprogramming, the OS decides which process will get memory when and how much.
- 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. 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.

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**.
- Decides who gets the resources.
- Allocates the resources.
- De-allocates the resources.

Other Important Activities

Following are some of the important activities that an Operating System performs –

- **Security** – By means of password and similar other techniques, it prevents 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.

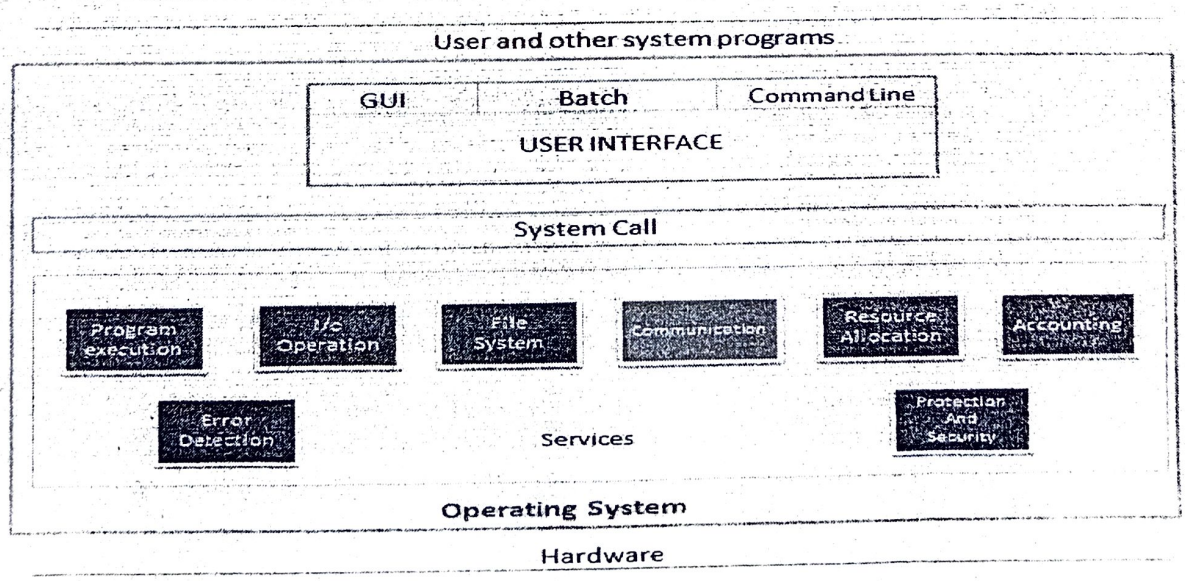
TOPIC-3: Operating System - Services

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 –

- 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.

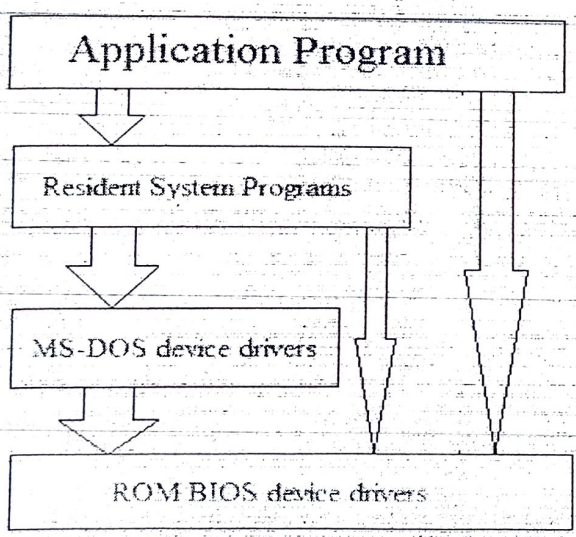
TOPIC4: Different approaches or Structures of Operating Systems

Operating system can be implemented with the help of various structures. The structure of the OS depends mainly on how the various common components of the operating system are interconnected and melded into the kernel. Depending on this we have following structures of the operating system:

1.Simplestructure:

Such operating systems do not have well defined structure and are small, simple and limited systems. The interfaces and levels of functionality are not well separated. MS-DOS is an example of such operating system. In MS-DOS application programs are able to access the basic I/O routines. These types of operating system cause the entire system to crash if one of the user programs fails.

Diagram of the structure of MS-DOS is shown below.



Advantages of Simple structure:

- It delivers better application performance because of the few interfaces between the application program and the hardware.
- Easy for kernel developers to develop such an operating system.

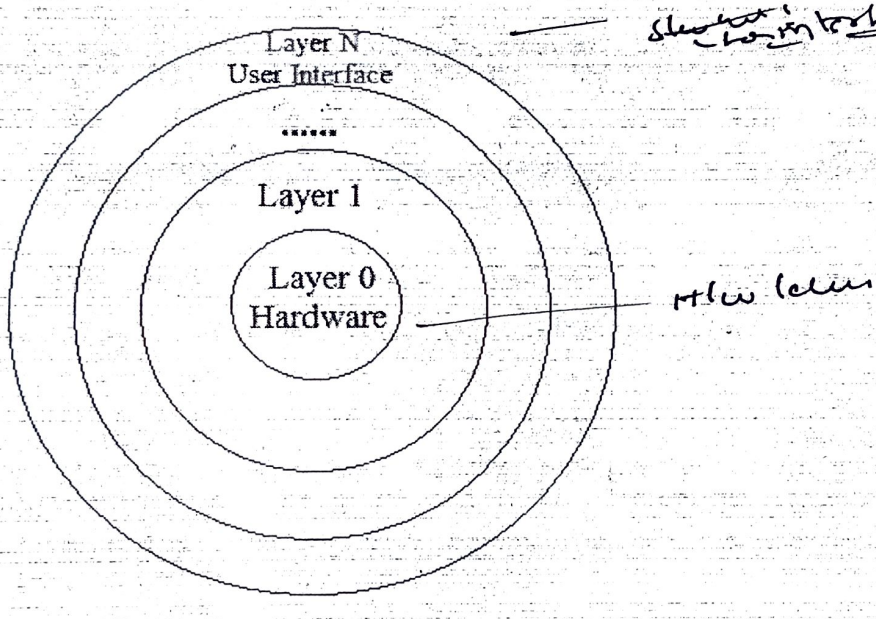
Disadvantages of Simple structure:

- The structure is very complicated as no clear boundaries exists between modules.
- It does not enforce data hiding in the operating system.

2.Layeredstructure:

An OS can be broken into pieces and retain much more control on system. In this structure the OS is broken into number of layers (levels). The bottom layer (layer 0) is the hardware and the topmost layer (layer N) is the user interface. These layers are so designed that each layer uses the functions of the lower level layers only. This simplifies the debugging process as if lower level layers are debugged and an error occurs during debugging then the error must be on that layer only as the lower level layers have already been debugged.

The main disadvantage of this structure is that at each layer, the data needs to be modified and passed on which adds overhead to the system. Moreover careful planning of the layers is necessary as a layer can use only lower level layers. UNIX is an example of this structure.



Advantages of Layered structure:

- Layering makes it easier to enhance the operating system as implementation of a layer can be changed easily without affecting the other layers.
- It is very easy to perform debugging and system verification.

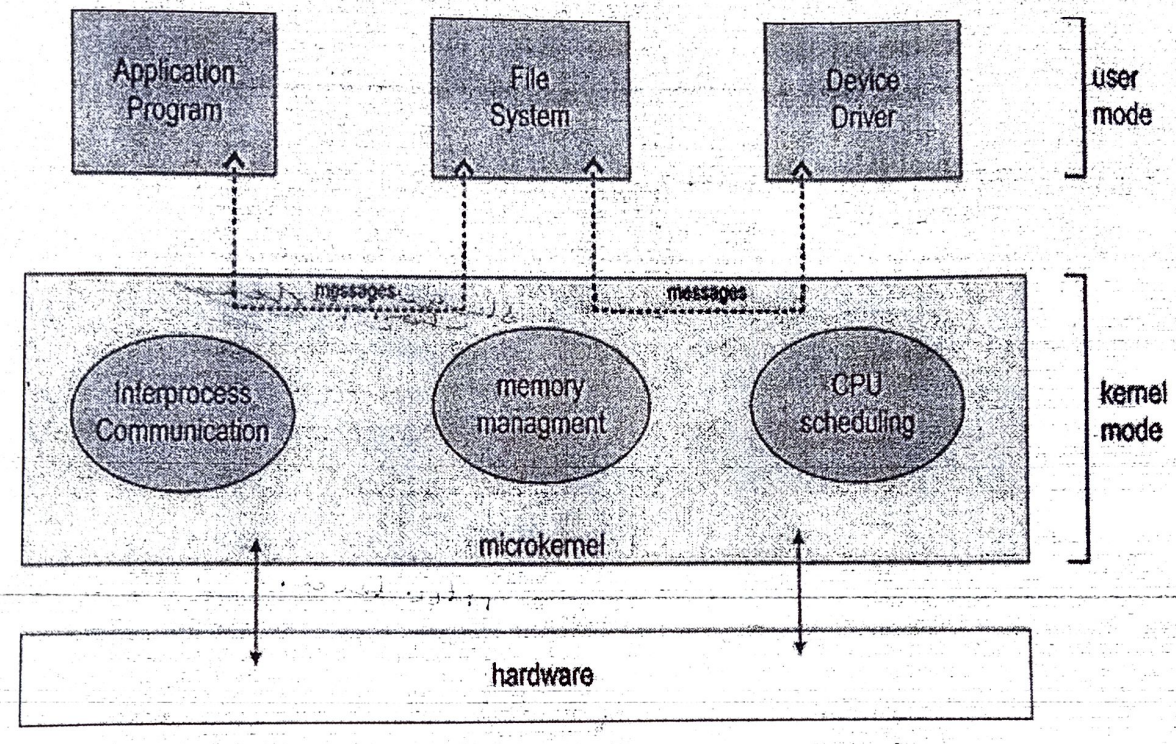
Disadvantages of Layered structure:

- In this structure the application performance is degraded as compared to simple structure.
- It requires careful planning for designing the layers as higher layers use the functionalities of only the lower layers.

3.Micro-kernel:

This structure designs the operating system by removing all non-essential components from the kernel and implementing them as system and user programs. This result in a smaller kernelcalledthemicro-kernel.

Advantages of this structure are that all new services need to be added to user space and does not require the kernel to be modified. Thus it is more secure and reliable as if a service fails then rest of the operating system remains untouched. Mac OS is an example of this type of OS.



Advantages of Micro-kernel structure:

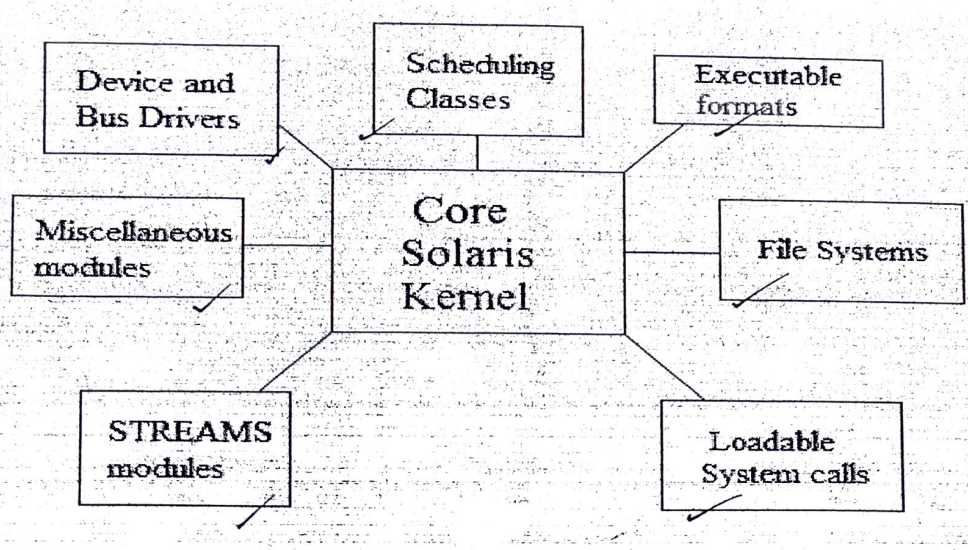
- It makes the operating system portable to various platforms.
- As microkernels are small so these can be tested effectively.

Disadvantages of Micro-kernel structure:

- Increased level of inter module communication degrades system performance.

4.Modularstructureorapproach:

It is considered as the best approach for an OS. It involves designing of a modular kernel. The kernel has only set of core components and other services are added as dynamically loadable modules to the kernel either during run time or boot time. It resembles layered structure due to the fact that each kernel has defined and protected interfaces but it is more flexible than the layered structure as a module can call any other module. For example Solaris OS is organized as shown in the figure.



5.Unix Operating System

Unix is a multiuser, multitasking operating system that was developed by Bell Laboratories in 1969. In a multiuser system, many users can use the system simultaneously. A multitasking system is capable of doing multiple jobs. Each user interacts with their own shell instance in this type of operating system and can start applications as required.

An image that demonstrates the structure of the Unix operating system is

As seen in the image, the main components of the Unix operating system structure are the kernel layer, the shell layer and the application layer.

Details about these are given as follows –

Kernel

The kernel provides a bridge between the hardware and the user. It is a software application that is central to the operating system. The kernel handles the files, memory, devices, processes and the network for the operating system. It is the responsibility of the kernel to make sure all the system and user tasks are performed correctly.

Shell

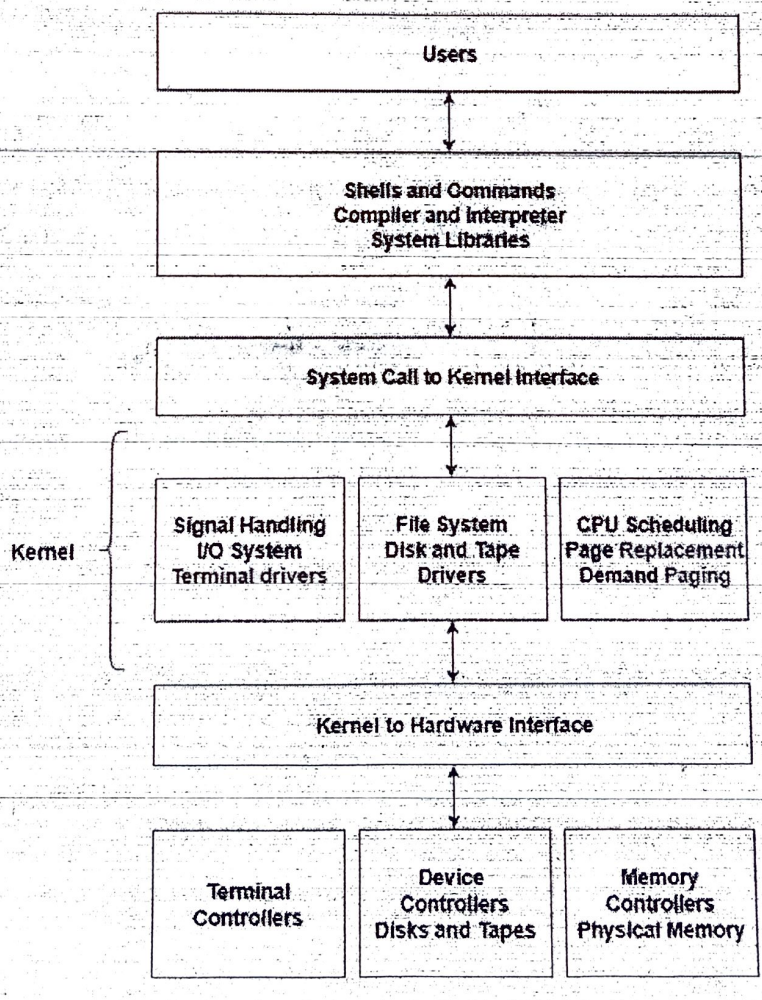
The program between the user and the kernel is known as the shell. It translates the many commands that are typed into the terminal session. These commands are known as the shell script. There are two major types of shells in Unix. These are Bourne shell and C Shell. The Bourne shell is the default shell for version 7 Unix.

The character $ is the default prompt for the Bourne shell. The C shell is a command processor that is run in a text window. The character % is the default prompt for the C shell.

Applications

The applications and utility layer in Unix includes the word processors, graphics programs, database management programs, commands etc. The application programs provide an application to the end users.

For example, a web browser is used to find information while gaming software is used to play games. The requests for service and application communication systems used in an application by a programmer is known as an application program interface (API).



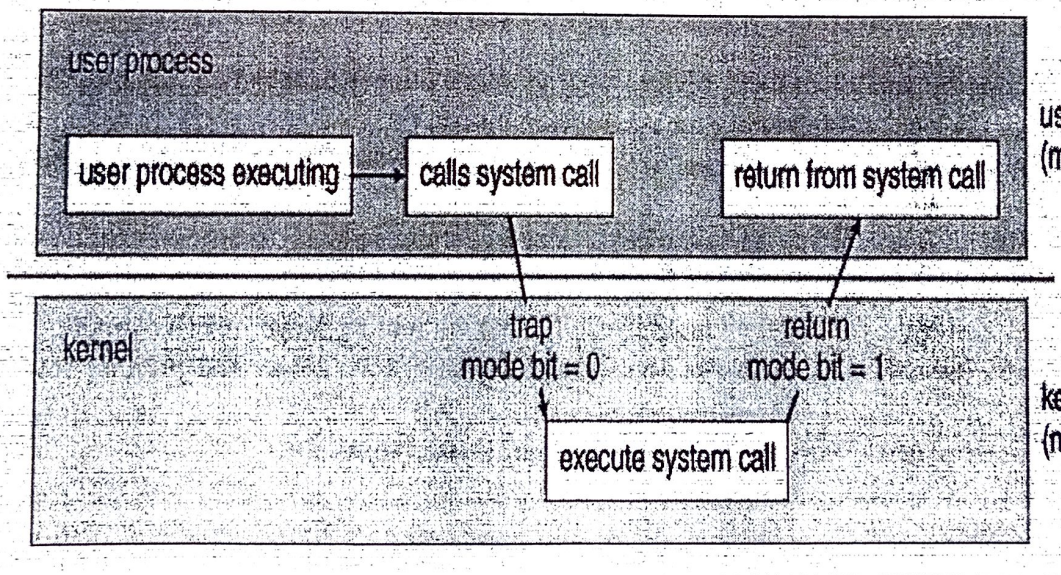
Unix Operating System Structure

TOPIC: 5Operating-System Operations

Interrupt-driven nature of modern OSes requires that erroneous processes not be able to disturb anything else

Dual-Mode and Multimode Operation

- User mode when executing harmless code in user applications
- Kernel mode (a.k.a. system mode, supervisor mode, privileged mode) when executing potentially dangerous code in the system kernel.
- Certain machine instructions (privileged instructions) can only be executed in kernel mode.
- Kernel mode can only be entered by making system calls. User code cannot flip the mode switch.
- Modern computers support dual-mode operation in hardware, and therefore most modern OSes support dual-mode operation.



- The concept of modes can be extended beyond two, requiring more than a single mode bit
- CPUs that support virtualization use one of these extra bits to indicate when the virtual machine manager, VMM, is in control of the system. The VMM has more privileges than ordinary user programs, but not so many as the full kernel.
- System calls are typically implemented in the form of software interrupts, which causes the hardware's interrupt handler to transfer control over to an appropriate interrupt handler, which is part of the operating system, switching the mode bit to kernel mode in the process. The interrupt handler checks exactly which interrupt was generated, checks additional parameters (generally passed through registers) if appropriate, and then calls the appropriate kernel service routine to handle the service requested by the system call.

- User programs' attempts to execute illegal instructions (privileged or non-existent instructions), or to access forbidden memory areas, also generate software interrupts, which are trapped by the interrupt handler and control is transferred to the OS, which issues an appropriate error message, possibly dumps data to a log (core) file for later analysis, and then terminates the offending program.

Timer

- Before the kernel begins executing user code, a timer is set to generate an interrupt.
- The timer interrupt handler reverts control back to the kernel.
- This assures that no user process can take over the system.
- Timer control is a privileged instruction, (requiring kernel mode.)

TOPIC: 6Computing Environments

Traditional Computing

Mobile Computing

- Computing on small handheld devices such as smart phones or tablets. (As opposed to laptops, which still fall under traditional computing.)
- May take advantage of additional built-in sensors, such as GPS, tilt, compass, and inertial movement.
- Typically connect to the Internet using wireless networking (IEEE 802.11) or cellular telephone technology.
- Limited in storage capacity, memory capacity, and computing power relative to a PC.
- Generally uses slower processors, that consume less battery power and produce less heat.
- The two dominant OSes today are Google Android and Apple iOS.

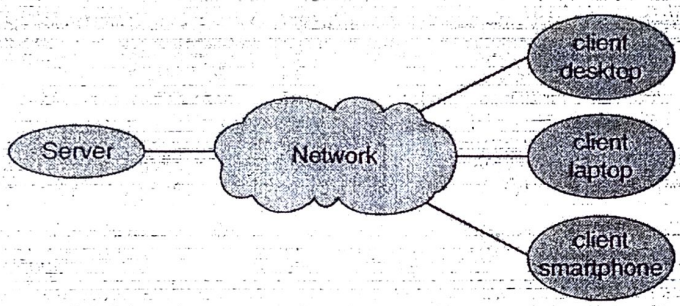
Distributed Systems

- Distributed Systems consist of multiple, possibly heterogeneous, computers connected together via a network and cooperating in some way, form, or fashion.
- Networks may range from small tight LANs to broad reaching WANs.
    - WAN = Wide Area Network, such as an international corporation
    - MAN =Metropolitan Area Network, covering a region the size of a city for example.
    - LAN =Local Area Network, typical of a home, business, single-site corporation, or university campus.
    - PAN = Personal Area Network, such as the bluetooth connection between your PC, phone, headset, car, etc.
- Network access speeds, throughputs, reliabilities, are all important issues.
- OS view of the network may range from just a special form of file access to complex well-coordinated network operating systems.

- Shared resources may include files, CPU cycles, RAM, printers, and other resources.

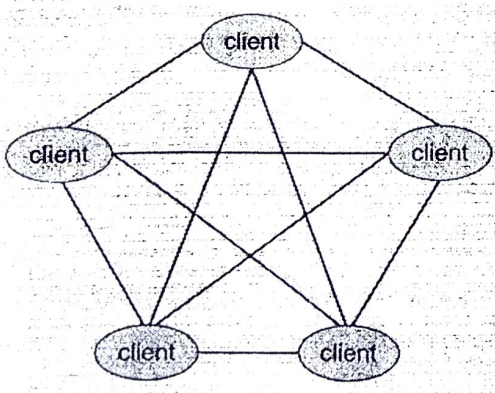
Client-Server Computing

- A defined server provides services (HW or SW) to other systems which serve as clients. (Technically clients and servers are processes, not HW, and may co-exist on the same physical computer.)
- A process may act as both client and server of either the same or different resources.
- Served resources may include disk space, CPU cycles, time of day, IP name information, graphical displays (X Servers), or other resources



Peer-to-Peer Computing

- Any computer or process on the network may provide services to any other which requests it. There is no clear "leader" or overall organization.
- May employ a central "directory" server for looking up the location of resources, or may use peer-to-peer searching to find resources.
- E.g. Skype uses a central server to locate a desired peer, and then further communication is peer to peer.

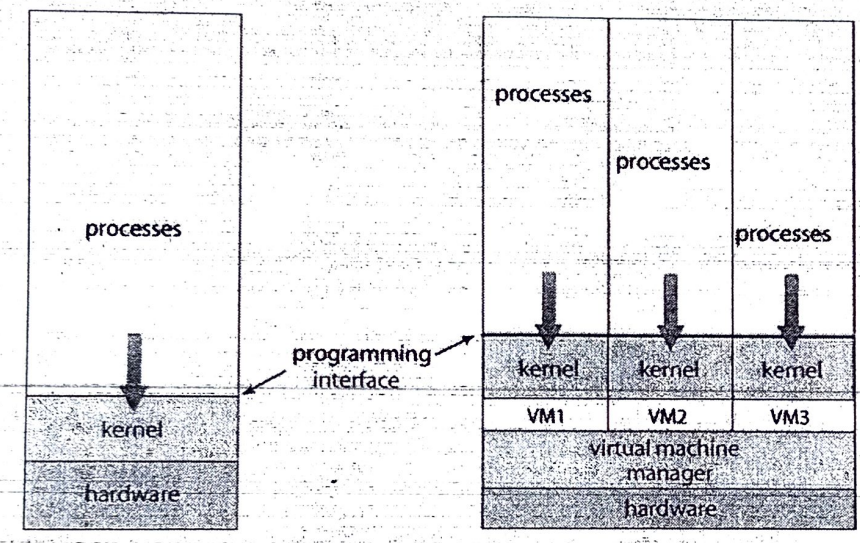


Virtualization

- Allows one or more "guest" operating systems to run on virtual machines hosted by a single physical machine and the virtual machine manager.
- Useful for cross-platform development and support.
- For example, a student could run UNIX on a virtual machine, hosted by a virtual machine manager on a Windows based personal computer. The student would have

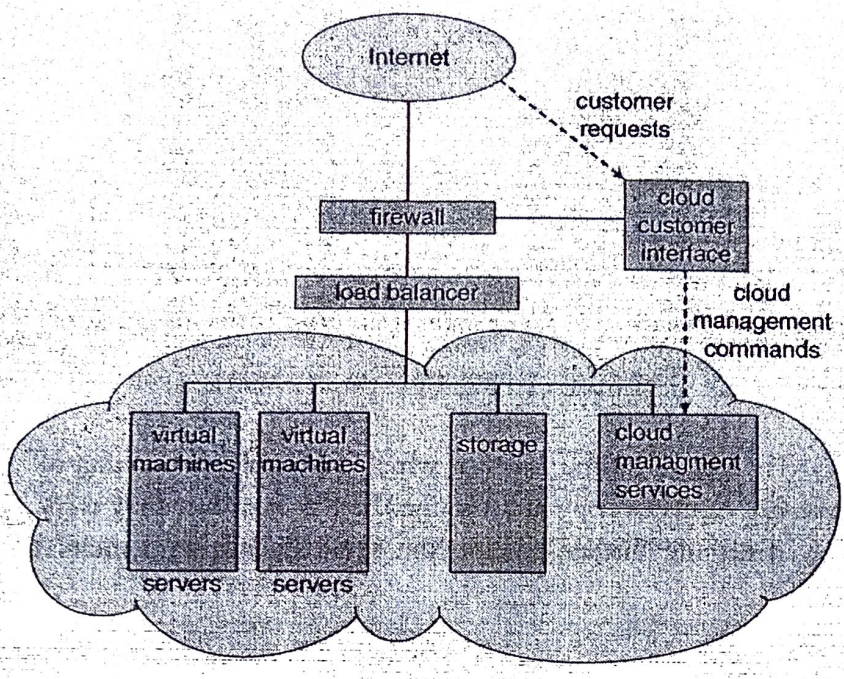
full root access to the virtual machine, and if it crashed, the underlying Windows machine should be unaffected.

- System calls have to be caught by the VMM and translated into (different) system calls made to the real underlying OS.
- Virtualization can slow down program that have to run through the VMM, but can also speed up some things if virtual hardware can be accessed through a cache instead of a physical device.
- Depending on the implementation, programs can also run simultaneously on the native OS, bypassing the virtual machines.



Cloud Computing

- Delivers computing, storage, and applications as a service over a network.
- Types of cloud computing:
  - Public cloud - Available to anyone willing to pay for the service.
  - Private cloud - Run by a company for internal use only.
  - Hybrid cloud - A cloud with both public and private components.
  - Software as a Service - SaaS - Applications such as word processors available via the Internet
  - Platform as a Service - PaaS - A software stack available for application use, such as a database server
  - Infrastructure as a Service - IaaS - Servers or storage available on the Internet, such as backup servers, photo storage, or file storage.
  - Service providers may provide more than one type of service
- Clouds may contain thousands of physical computers, millions of virtual ones, and petabytes of total storage.
- Web hosting services may offer (one or more) virtual machine(s) to each of their clients.



Real-Time Embedded Systems

- Embedded into devices such as automobiles, climate control systems, process control, and even toasters and refrigerators.
- May involve specialized chips, or generic CPUs applied to a particular task. (Consider the current price of 80286 or even 8086 or 8088 chips, which are still plenty powerful enough for simple electronic devices such as kids toys.)
- Process control devices require real-time (interrupt driven) OSes. Response time can be critical for many such devices.

TOPIC:7:Open-Source Operating Systems

- (For more information on the Flourish conference held at UIC on the subject of Free Libre and Open Source Software, visit http://www.flourishconf.com)
- Open-Source software is published (sometimes sold) with the source code, so that anyone can see and optionally modify the code.
- Open-source SW is often developed and maintained by a small army of loosely connected often unpaid programmers, each working towards the common good.
- Critics argue that open-source SW can be buggy, but proponents counter that bugs are found and fixed quickly, since there are so many pairs of eyes inspecting all the code.
- Open-source operating systems are a good resource for studying OS development, since students can examine the source code and even change it and re-compile the changes.

History

- At one time (1950s) a lot of code was open-source.
- Later, companies tried to protect the privacy of their code, particularly sensitive issues such as copyright protection algorithms.
- In 1983 Richard Stallman started the GNU project to produce an open-source UNIX.
- He later published the GNU Manifesto, arguing that ALL software should be open-source, and founded the Free Software Foundation to promote open-source development.
- FSF and GNU use the GNU General Public License which essentially states that all users of the software have full rights to copy and change the SW however they wish, so long as anything they distribute further contain the same license agreement. (Copylefting)

Linux

- Developed by Linus Torvalds in Finland in 1991 as the first full operating system developed by GNU.
- Many different distributions of Linux have evolved from Linus's original, including RedHat, SUSE, Fedora, Debian, Slackware, and Ubuntu, each geared toward a different group of end-users and operating environments.
- To run Linux on a Windows system using VMware, follow these steps:
  1. Download the free "VMware Player" tool from http://www.vmware.com/download/player and install it on your system
  2. Choose a Linux version from among hundreds of virtual machine images at http://www.vmware.com/appliances
  3. Boot the virtual machine within VMware Player.

BSD UNIX

- UNIX was originally developed at ATT Bell labs, and the source code made available to computer science students at many universities, including the University of California at Berkeley, UCB.
- UCB students developed UNIX further, and released their product as BSD UNIX in both binary and source-code format.
- BSD UNIX is not open-source, however, because a license is still needed from ATT.
- In spite of various lawsuits, there are now several versions of BSD UNIX, including Free BSD, NetBSD, OpenBSD, and DragonflyBSD
- The source code is located in /usr/src.
- The core of the Mac operating system is Darwin, derived from BSD UNIX, and is available at http://developer.apple.com/opensource/index.html

Solaris

- Solaris is the UNIX operating system for computers from Sun Microsystems.
- Solaris was originally based on BSD UNIX, and has since migrated to ATT SystemV as its basis.
- Parts of Solaris are now open-source, and some are not because they are still covered by ATT copyrights.
- It is possible to change the open-source components of Solaris, re-compile them, and then link them in with binary libraries of the copyrighted portions of Solaris.
- Open Solaris is available from http://www.opensolaris.org/os/
- Solaris also allows viewing of the source code online, without having to download and unpack the entire package.

Utility

- The free software movement is gaining rapidly in popularity, leading to thousands of ongoing projects involving untold numbers of programmers.
- Sites such as http://freshmeat.net/ and http://distrowatch.com/ provide portals to many of these projects.
- **Cosmos**
- This is an open source operating system written mostly in programming language C#. Its full form is C# Open Source Managed Operating System. Till 2016, Cosmos did not intend to be a fully fledged operating system but a system that allowed other developers to easily build their own operating systems. It also hid the inner workings of the hardware from the developers thus providing data abstraction.
- **FreeDOS**
- This was a free operating system developed for systems compatible with IBM PC computers. FreeDOS provides a complete environment to run legacy software and other embedded systems. It can booted from a floppy disk or USB flash drive as required. FreeDos is licensed under the GNU General Public license and contains free and open source software. So there is no license fees required for its distribution and changes to the system are permitted.
- **Genode**
- Genode is free as well as open source. It contains a microkernel layer and different user components. It is one of the few open source operating systems not derived from a licenced operating system such as Unix. Genode can be used as an operating system for computers, tablets, etc. as required. It is also used as a base for virtualisation, interprocess communication, software development etc. as it has a small code system.
- **Ghost OS**
- This is a free, open source operating system developed for personal computers. It started as a research project and developed to contain various advanced features like graphical user interface, C library etc. The Ghost operating system features multiprocessing and multitasking and is based on the Ghost Kernel. Most of the programming in Ghost OS is done in C++.

- **ITS**
- The incompatible time-sharing system was developed by the MIT Artificial Intelligence Library. It is principally a time sharing system. There is a remote login facility which allowed guest users to informally try out the operating system and its features using ARPAnet. ITS also gave out many new features that were unique at that time such as device independent graphics terminal, virtual devices, inter machine file system access etc.
- **OSv**
- This was an operating system released in 2013. It was mainly focused on cloud computing and was built to run on top of a virtual machine as a guest. This is the reason it doesn't include drivers for bare hardware. In the OSv operating system, everything runs in the kernel address space and there is no concept of a multi-user system.
- **Phantom OS**
- This is an operating system that is based on the concepts on persistent virtual memory and is code oriented. It was mostly developed by Russian developers. Phantom OS is not based on concepts of famous operating systems such as Unix. Its main goal is simplicity and effectiveness in process management

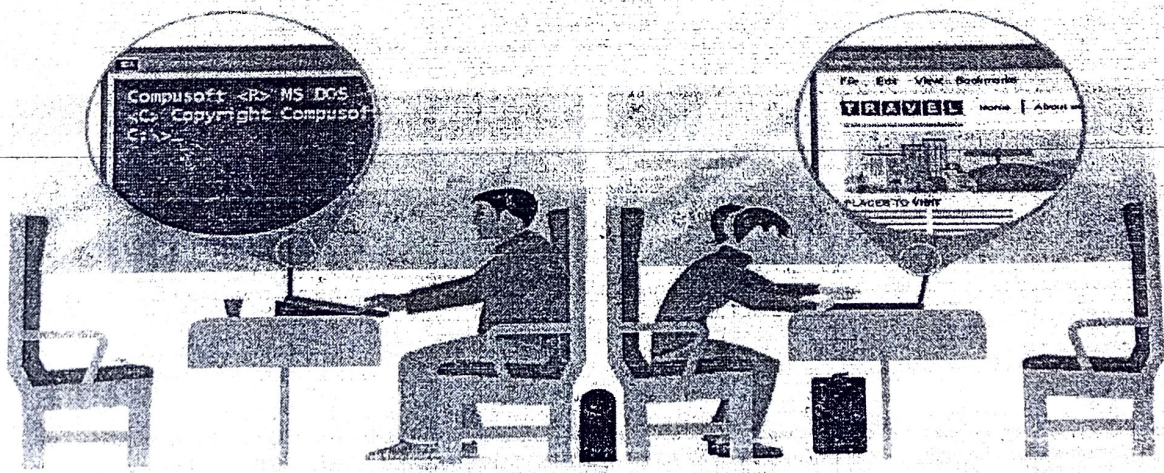
TOPIC 8: User and Operating-System Interface

Operating systems

An operating system is software that manages computer hardware and software. It supplies an interface for the user and important utilities for managing the computer

User interface

The OS provides a user interface (UI), an environment for the user to interact with the machine. The UI is either graphical or text-based.



Graphical user interface (GUI)

The OS on most computers and smartphones provides an environment with tiles, icons and/or menus. This type of interface is called the graphical user interface (GUI) because the user interacts with images through a mouse, keyboard or touchscreen.

Command line interface (CLI)

An OS also provides a method of interaction that is non-graphical, called the command line interface (CLI). This is a text-only service with feedback from the OS appearing in text. Using a CLI requires knowledge of the commands available on a particular machine.

Advantages of using the command line include:

- a **faster** way to get tasks done
- it is **more flexible** than a GUI
- it uses **less** memory

Some games, such as Minecraft, also make use of a command line tool which allows the user to bypass the main interface and alter the game's mechanics or environment.

TOPIC: 9 SYSTEM CALLS IN OS:

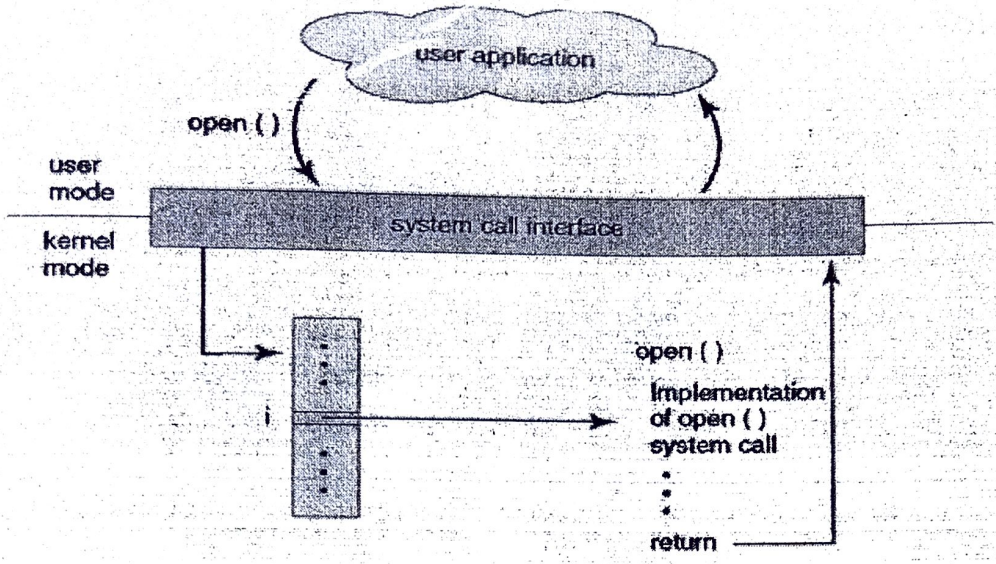
□ System calls provide an interface between the process and the operating system.

□ System calls allow user-level processes to request some services from the operating system which process itself is not allowed to do.

□ In handling the trap, the operating system will enter in the kernel mode, where it has access to privileged instructions, and can perform the desired service on the behalf of user-level process.

□ It is because of the critical nature of operations that the operating system itself does them every time they are needed.

□ For example, for I/O a process involves a system call telling the operating system to read or write particular area and this request is satisfied by the operating system.



TOPIC: 10 Types of System calls

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

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.
- Following are functions of process control:

i. end, abort

ii. load, execute

iii. create process, terminate process

iv. get process attributes, set process attributes

v. wait for time

vi. wait event, signal event

vii. allocate and free memory

2) File management :

- We first need to be able to create and delete files. Either system call requires the name of the file and perhaps some of the file's attributes.
- Once the file is created, we need to open it and to use it. We may also read, write, or reposition. Finally, we need to close the file, indicating that we are no longer using it.
- We may need these same sets of operations for directories if we have a directory structure for organizing files in the file system.
- In addition, for either files or directories, we need to be able to determine the values of various attributes and perhaps to reset them if necessary. File attributes include the file name, a file type, protection codes, accounting information, and so on

Functions:

- create file, delete file
- open, close file
- read, write, reposition
- get and set file attributes

3) Device Management:

- A process may need several resources to execute - main memory, disk drives, access to files, and so on. If the resources are available, they can be granted, and control can be returned to the user process. Otherwise, the process will have to wait until sufficient resources are available.
- The various resources controlled by the OS can be thought of as devices. Some of these devices are physical devices (for example, tapes), while others can be thought of as abstract or virtual devices (for example, files).
- Once the device has been requested (and allocated to us), we can read, write, and (possibly) reposition the device, just as we can with files.
- In fact, the similarity between I/O devices and files is so great that many OSs, including UNIX, merge the two into a combined file-device structure.
- A set of system calls is used on files and devices. Sometimes, I/0 devices are identified by special file names, directory placement, or file attributes.

Functions:

- request device, release device
- read, write, reposition
- get device attributes, set device attributes
- logically attach or detach devices

Information Maintenance

- Many system calls exist simply for the purpose of transferring information between the user program and the OS. For example, most systems have a system call to return the current time and date.
- Other system calls may return information about the system, such as the number of current users, the version number of the OS, the amount of free memory or disk space, and so on.
- In addition, the OS keeps information about all its processes, and system calls are used to access this information. Generally, calls are also used to reset the process information.

Functions:

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

Communication

- There are two common models of interprocess communication: the message-passing model and the shared-memory model. In the message-passing model, the communicating processes exchange messages with one another to transfer information.
- In the shared-memory model, processes use shared memory creates and shared memory attaches system calls to create and gain access to regions of memory owned by other processes.
- Recall that, normally, the OS tries to prevent one process from accessing another process's memory. Shared memory requires that two or more processes agree to remove this restriction. They can then exchange information by reading and writing data in the shared areas.
- Message passing is useful for exchanging smaller amounts of data, because no conflicts need be avoided. It is also easier to implement than is shared memory for intercomputer communication.
- Shared memory allows maximum speed and convenience of communication, since it can be done at memory speeds when it takes place within a computer. Problems exist, however, in the areas of protection and synchronization between the processes sharing memory.

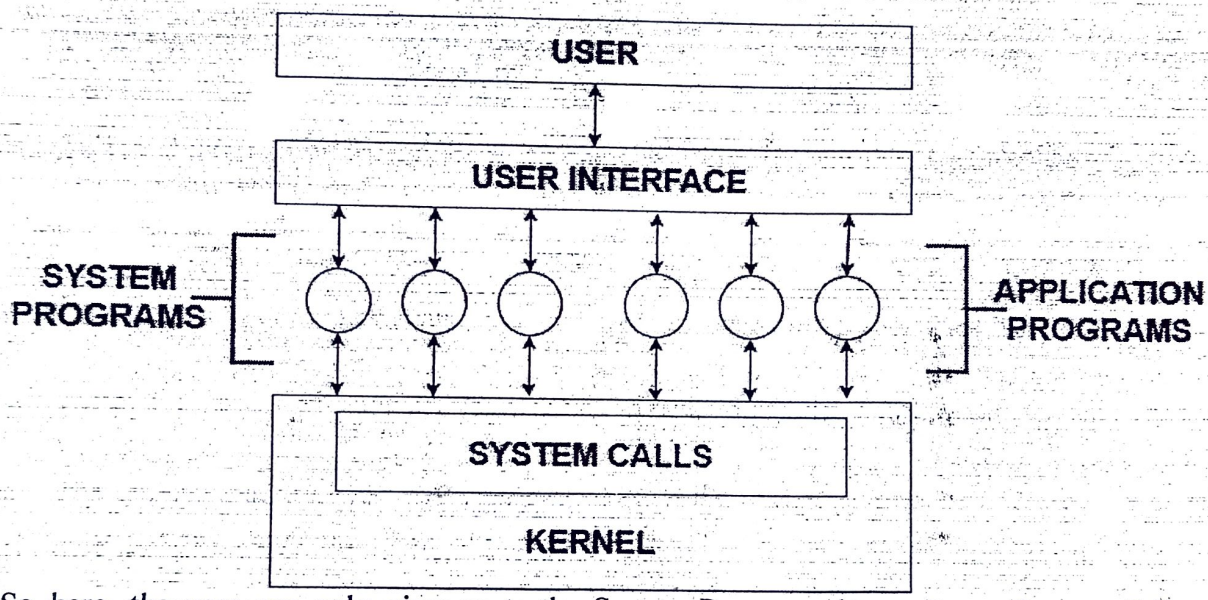
Functions:

  - create, delete communication connection
  - send, receive messages
  - transfer status information

- Attach and Detach remote devices

TOPIC:11:System Programs in Operating System

System Programming can be defined as the act of building Systems Software using System Programming Languages. According to Computer Hierarchy, one which comes at last is Hardware. Then it is Operating System, System Programs, and finally Application Programs. Program Development and Execution can be done conveniently in System Programs. Some of the System Programs are simply user interfaces, others are complex. It traditionally lies between the user interface and system calls.



So here, the user can only view up-to-the System Programs he can't see System Calls. System Programs can be divided into these categories :

1. **FileManagement–**
   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 placing 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. **StatusInformation**
   Information like date, time amount of available memory, or disk space is asked by some users. Others providing 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. **FileModification–**
   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 perform transformations of files we use special commands.

4. **Programming-Languagesupport–**
   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. **ProgramLoadingandExecution–**
   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, User can send e-mail, browsing on web pages, remote login, the transformation of files from one user toanother. Some examples of system program in O.S. are –

- Windows 10
- Mac OS X
- Ubuntu

- Linux
- Unix
- Android
- Anti-virus
- Disk formatting
- Computer language translators

TOPIC:13 Operating System Debugging

Debugging is the process of finding the problems in a computer system and solving them. There are many different ways in which operating systems perform debugging. Some of these are –

Log Files

The log files record all the events that occur in an operating system. This is done by writing all the messages into a log file. There are different types of log files. Some of these are given as follows –

Event Logs

These stores the records of all the events that occur in the execution of a system. This is done so that the activities of all the events can be understood to diagnose problems.

Transaction Logs

The transaction logs store the changes to the data so that the system can recover from crashes and other errors. These logs are readable by a human.

Message Logs

These logs store both the public and private messages between the users. They are mostly plain text files, but in some cases they may be HTML files.

Core Dump Files

The core dump files contain the memory address space of a process that terminates unexpectedly. The creation of the core dump is triggered in response to program crashes by the kernel. The core dump files are used by the developers to find the program's state at the time of its termination so that they can find out why the termination occurred.

The automatic creation of the core dump files can be disabled by the users. This may be done to improve performance, clear disk space or increase security.

Crash Dump Files

In the event of a total system failure, the information about the state of the operating system is captured in crash dump files. There are three types of dump that can be captured when a system crashes. These are –

Complete Memory Dump

The whole contents of the physical memory at the time of the system crash are captured in the complete memory dump. This is the default setting on the Windows Server System.

Kernel Memory Dump

Only the kernel mode read and write pages that are present in the main memory at the time of the system crash are stored in the kernel memory dump.

Small Memory Dump

This memory dump contains the list of device drivers, stop code, process and thread information, kernel stack etc.

Trace Listings

The trace listing record information about a program execution using logging. This information is used by programmers for debugging. System administrators and technical personnel can use the trace listings to find the common problems with software using software monitoring tools.

Profiling

This is a type of program analysis that measures various parameters in a program such as space and time complexity, frequency and duration of function calls, usage of specific instructions etc. Profiling is done by monitoring the source code of the required system program using a code profiler.

TOPIC:14:Booting in Operating System

Booting is the process of starting a computer. It can be initiated by hardware such as a button press or by a software command. After it is switched on, a CPU has no software in its main memory, so some processes must load software into memory before execution. This may be done by hardware or firmware in the CPU or by a separate processor in the computer system.

Restarting a computer also is called rebooting, which can be "***hard***", e.g., after electrical power to the CPU is switched from off to on, or "***soft***", where the power is not cut. On some

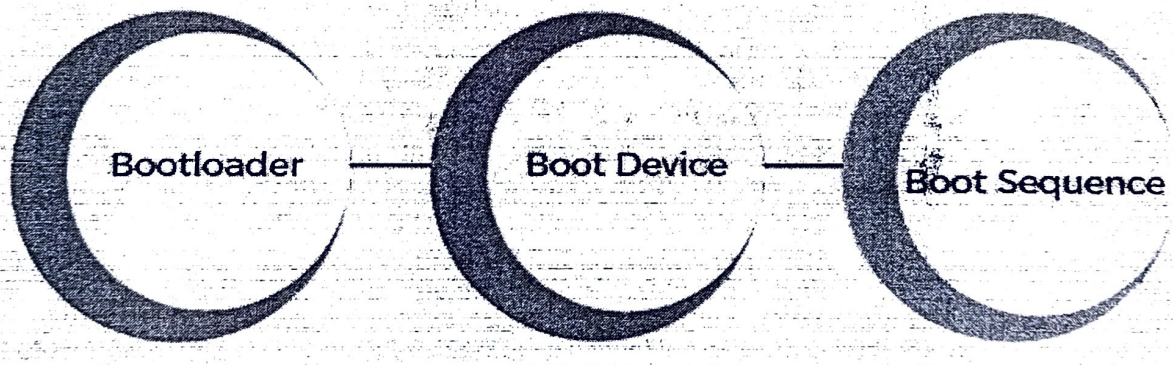
systems, a soft boot may optionally clear RAM to zero. Hard and soft booting can be initiated by hardware such as a button press or a software command. Booting is complete when the operative runtime system, typically the operating system and some applications, is attained.

The process of returning a computer from a state of sleep does not involve booting; however, restoring it from a state of hibernation does. Minimally, some embedded systems do not require a noticeable boot sequence to begin functioning and, when turned on, may run operational programs that are stored in ROM. All computer systems are state machines and a reboot may be the only method to return to a designated zero-state from an unintended, locked state.

In addition to loading an operating system or stand-alone utility, the boot process can also load a storage dump program for diagnosing problems in an operating system.

Sequencing of Booting

Booting is a start-up sequence that starts the operating system of a computer when it is turned on. A boot sequence is the initial set of operations that the computer performs when it is switched on. Every computer has a boot sequence.



1. Boot Loader: Computers powered by the central processing unit can only execute code found in the system's memory. Modern operating systems and application program code and data are stored on nonvolatile memories. When a computer is first powered on, it must initially rely only on the code and data stored in nonvolatile portions of the system's memory. The operating system is not really loaded at boot time, and the computer's hardware cannot perform many complex systems actions.

The program that starts the chain reaction that ends with the entire operating system being loaded is the boot loader or bootstrap loader. The boot loader's only job is to load other software for the operating system to start.

2. Boot Devices: The boot device is the device from which the operating system is loaded. A modern PC BIOS (Basic Input/Output System) supports booting from various devices. These include the local hard disk drive, optical drive, floppy drive, a network interface card, and a

USB device. The BIOS will allow the user to configure a boot order. If the boot order is set to:

- CD Drive
- Hard Disk Drive
- Network

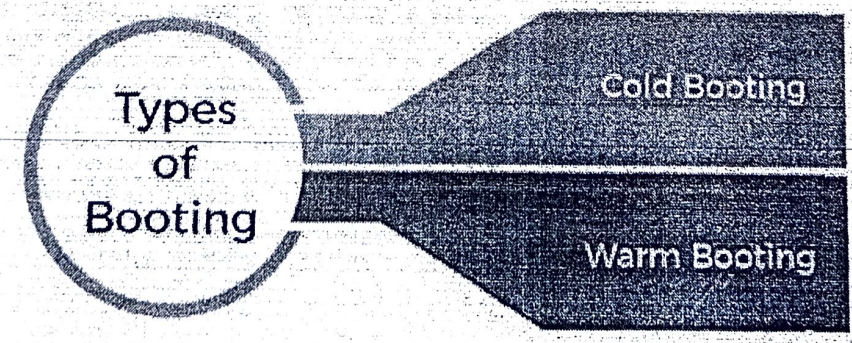
The BIOS will try to boot from the CD drive first, and if that fails, then it will try to boot from the hard disk drive, and if that fails, then it will try to boot from the network, and if that fails, then it won't boot at all.

3. Boot Sequence: There is a standard boot sequence that all personal computers use. First, the CPU runs an instruction in memory for the BIOS. That instruction contains a jump instruction that transfers to the BIOS start-up program. This program runs a power-on self-test (POST) to check that devices the computer will rely on are functioning properly. Then, the BIOS goes through the configured boot sequence until it finds a bootable device. Once BIOS has found a bootable device, BIOS loads the bootsector and transfers execution to the boot sector. If the boot device is a hard drive, it will be a master boot record (MBR).

The MBR code checks the partition table for an active partition. If one is found, the MBR code loads that partition's boot sector and executes it. The boot sector is often operating system specific, and however, in most operating systems, its main function is to load and execute the operating system kernel, which continues start-up. Suppose there is no active partition, or the active partition's boot sector is invalid. In that case, the MBR may load a secondary boot loader which will select a partition and load its boot sector, which usually loads the corresponding operating system kernel.

Types of Booting

There are two types of booting in an operating system.



1. **Cold Booting:** When the computer starts for the first time or is in a shut-down state and switch on the power button to start the system, this type of process to start the computer is called cold booting. During cold booting, the system will read all the instructions from the

ROM (BIOS) and the Operating System will be automatically get loaded into the system. This booting takes more time than Hot or Warm Booting.

2. **Warm Booting:** Warm or Hot Booting process is when computer systems come to no response or hang state, and then the system is allowed to restart during on condition. It is also referred to as rebooting. There are many reasons for this state, and the only solution is to reboot the computer. Rebooting may be required when we install new software or hardware. The system requires a reboot to set software or hardware configuration changes, or sometimes systems may behave abnormally or may not respond properly. In such a case, the system has to be a force restart. Most commonly ***Ctrl+Alt+Del*** button is used to reboot the system. Else, in some systems, the external reset button may be available to reboot the system.

Booting Process in Operating System

When our computer is switched on, it can be started by hardware such as a button press, or by software command, a computer's central processing unit (CPU) has no software in its main memory, there is some process which must load software into main memory before it can be executed. Below are the six steps to describe the boot process in the operating system, such as:

Step 1: Once the computer system is turned on, ***BIOS*** (Basic Input /Output System) performs a series of activities or functionality tests on programs stored in ROM, called on ***POST*** (Power-on Self Test) that checks to see whether peripherals in the system are in perfect order or not.

Step 2: After the BIOS is done with pre-boot activities or functionality test, it read bootable sequence from ***CMOS*** (Common Metal Oxide Semiconductor) and looks for master boot record in the first physical sector of the bootable disk as per boot device sequence specified in ***CMOS***. For example, if the boot device sequence is:

- Floppy Disk
- Hard Disk
- CDROM

Step 3: After this, the master boot record will search first in a floppy disk drive. If not found, then the hard disk drive will search for the master boot record. But if the master boot record is not even present on the hard disk, then the CDROM drive will search. If the system cannot read the master boot record from any of these sources, ROM displays "***No Boot device found***" and halted the system. On finding the master boot record from a particular bootable disk drive, the operating system loader, also called Bootstrap loader, is loaded from the boot

sector of that bootable drive into memory. A bootstrap loader is a special program that is present in the boot sector of a bootable drive.

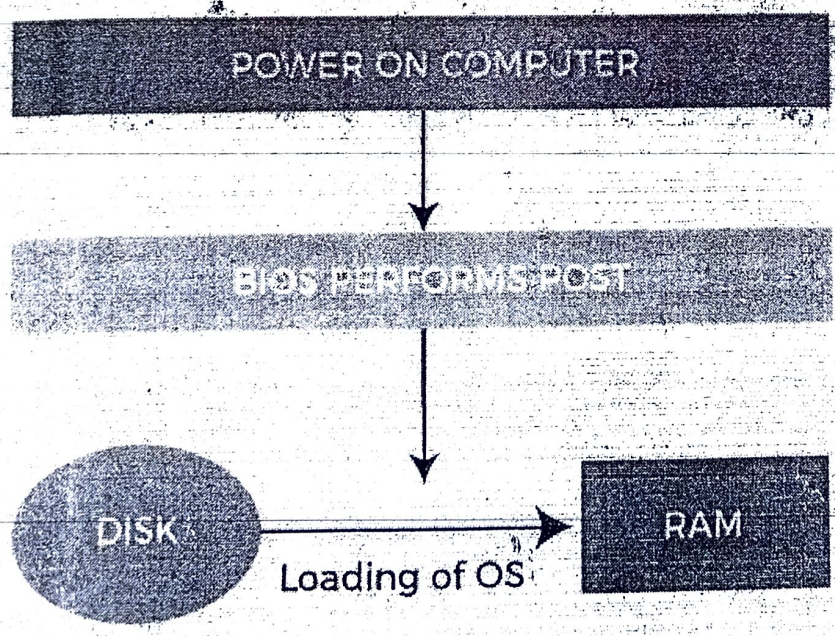
Step 4: The bootstrap loader first loads the ***IO.SYS*** file. After this, ***MSDOS.SYS*** file is loaded, which is the core file of the DOS operating system.

Step 5: After this, ***MSDOS.SYS*** file searches to find Command Interpreter in ***CONFIG.SYS*** file, and when it finds, it loads into memory. If no Command Interpreter is specified in the ***CONFIG.SYS*** file, the ***COMMAND.COM*** file is loaded as the default Command Interpreter of the DOS operating system.

Step 6: The last file is to be loaded and executed is the ***AUTOEXEC.BAT*** file that contains a sequence of DOS commands. After this, the prompt is displayed. We can see the drive letter of bootable drive displayed on the computer system, which indicates that the operating system has been successfully on the system from that drive.

What is Dual Booting

When two operating systems are installed on the computer system, then it is called dual booting. Multiple operating systems can be installed on such a system. But to know which operating system is to boot, a boot loader that understands multiple file systems and multiple operating systems can occupy the boot space.



Once loaded, it can boot one of the operating systems available on the disk. The disk can have multiple partitions, each containing a different type of operating system. When a computer system turns on, a boot manager program displays a menu, allowing the user to choose the operating system to use.

Operating Systems Client/Server Communication

Client/Server communication involves two components, namely a client and a server. They are usually multiple clients in communication with a single server. The clients send requests to the server and the server responds to the client requests.

There are three main methods to client/server communication. These are given as follows –

Sockets

Sockets facilitate communication between two processes on the same machine or different machines. They are used in a client/server framework and consist of the IP address and port number. Many application protocols use sockets for data connection and data transfer between a client and a server.

Socket communication is quite low-level as sockets only transfer an unstructured byte stream across processes. The structure on the byte stream is imposed by the client and server applications.

A diagram that illustrates sockets is as follows –

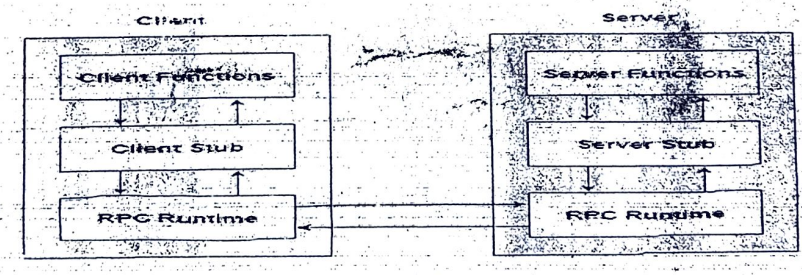
A diagram that illustrates sockets is as follows –

Remote Procedure Calls

These are interprocess communication techniques that are used for client-server based applications. A remote procedure call is also known as a subroutine call or a function call.

A client has a request that the RPC translates and sends to the server. This request may be a procedure or a function call to a remote server. When the server receives the request, it sends the required response back to the client.

A diagram that illustrates remote procedure calls is given as follows –



Pipes

These are interprocess communication methods that contain two end points. Data is entered from one end of the pipe by a process and consumed from the other end by the other process.

The two different types of pipes are ordinary pipes and named pipes. Ordinary pipes only allow one way communication. For two way communication, two pipes are required. Ordinary pipes have a parent child relationship between the processes as the pipes can only be accessed by processes that created or inherited them.

Named pipes are more powerful than ordinary pipes and allow two way communication. These pipes exist even after the processes using them have terminated. They need to be explicitly deleted when not required anymore.

A diagram that demonstrates pipes are given as follows –