OPERATING SYSTEMS

MODULE-1

BCS303

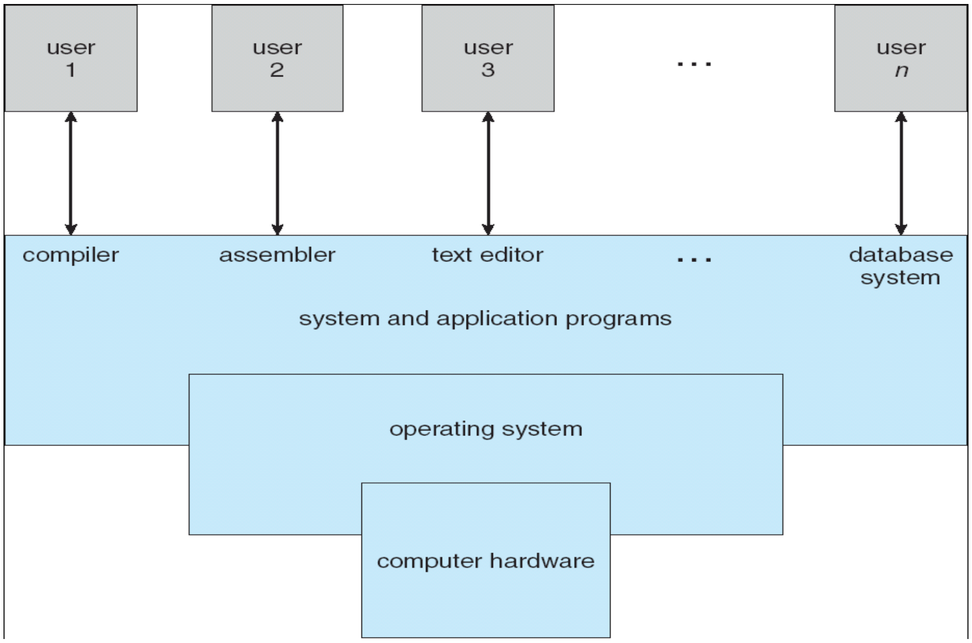
INTRODUCTION TO OPERATING SYSTEMS

What is an Operating System?

An **operating system** is a program that manages the computer hardware, and acts as an intermediary between the computer user and the computer hardware. It also provides a basis for application programs.

WHAT OPERATING SYSTEMS DO

The operating system's role in the overall computer system is studied, by considering a computer system which can be divided roughly into four components: the *hardware,* the *operating system*, the *application programs*, and the *users.*



- Hardware provides basic computing resources such as CPU, memory, I/O devices
- System & Application programs define the ways in which the system resources are used to solve the computing problems of the users / systems. Word processors, compilers, web browsers, database systems, video games.
- Operating system controls and coordinates use of hardware among various applications and users
- Users: People, machines, other computers.

VIEWS OF OPERATING SYSTEMS

**********Explain different views of operating systems.**

There are two viewpoints of Operating system

1. User view
2. System View

User View of computer system varies according to the interface being used.

- ✓ **PC**: In this system the operating system is designed mostly for ease of **use.**
- ✓ **Mainframe or Minicomputer:** Operating system is designed to maximize resource utilization.
- ✓ **Workstation:** Operating system is designed to compromise between individual usability and resource utilization.
- ✓ **Handheld devices**: Their operating systems are designed mostly for individual usability, but performance per amount of battery life is important as well.
- ✓ **Embedded computers in home devices and automobiles**: 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

i. **Resource allocator:**
   - **a)** Manages all resources
   - **b)** Decides between conflicting requests for efficient and fair resource use

ii. **Control program:**
   - a) Control the various I/O devices and user programs.
   - b) Controls execution of programs to prevent errors and improper use of the computer

DEFINING OPERATING SYSTEMS

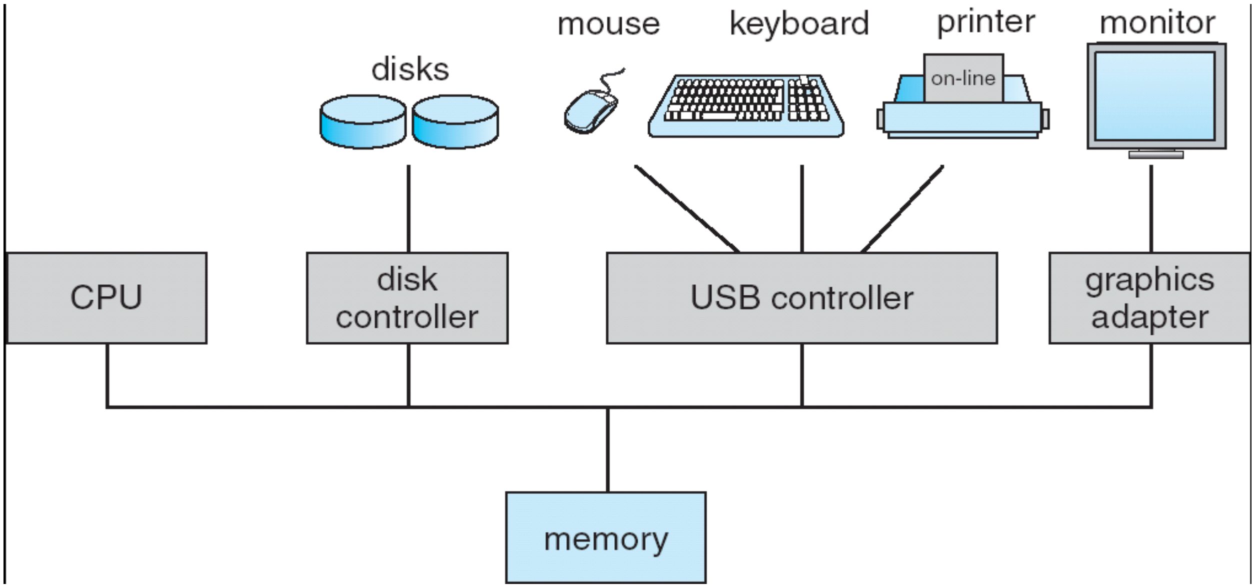
- The common functions of controlling and allocating resources are brought together in one piece of software called Operating System
- "The one program running at all times on the computer" **is the kernel**. Everything else is either a system program (ships with the operating system) or an application program.

COMPUTER-SYSTEM ORGANIZATION

1. Computer-system operation

a. One or more CPUs, device controllers connect through common bus providing access to shared memory.

b. Concurrent execution of CPUs and devices competing for memory cycles.

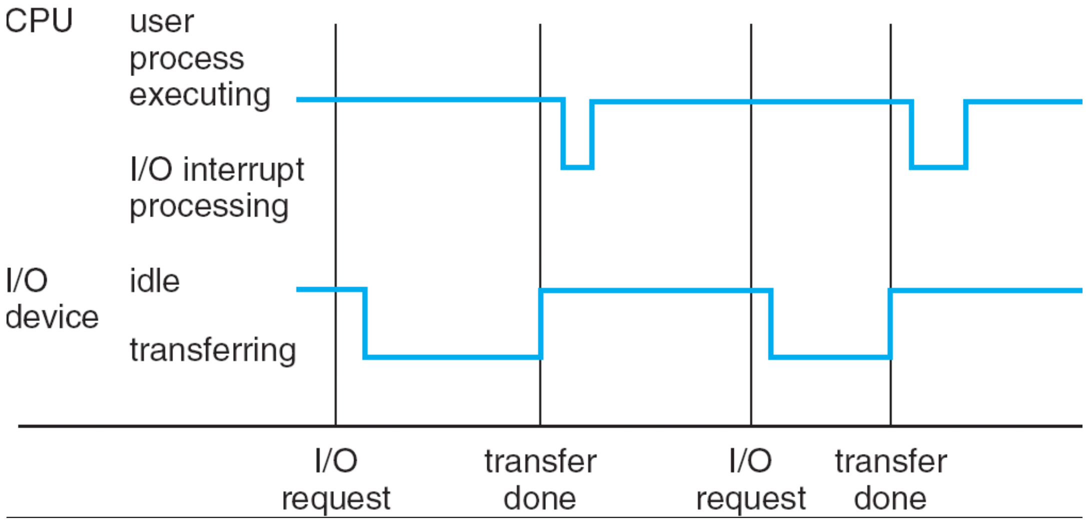


Bootstrap program: For a computer to start running—for instance, when it is powered up or rebooted—it needs to have an initial program to run. This initial program, or **bootstrap program,** tends to be simple.

- ✓ Typically, it is stored in read-only memory (ROM) or electrically erasable programmable read-only memory (EEPROM), known by the general term **firmware,** within the computer hardware.
- ✓ It initializes all aspects of the system (CPU registers, Device Controllers, memory contents)
- ✓ Loads operating system kernel and starts execution.
- ✓ The operating system then starts executing the first process, such as "init," and waits for some event to occur.
- ✓ The occurrence of an event is usually signaled by an **interrupt** from either the hardware or the software.

Interrupt

- ✓ The occurrence of event is signaled by interrupt from either hardware(signal to CPU) or software (system call)
- ✓ Interrupt transfers control to the interrupt service routine generally, through the *interrupt vector*, which contains the addresses of all the service routines.
- ✓ Interrupt architecture must save the address of the interrupted instruction.
- ✓ Incoming interrupts are *disabled* while another interrupt is being processed to prevent a *lost interrupt*.
- ✓ A *trap* is a software-generated interrupt caused either by an error or a user request.
- ✓ An operating system is *interrupt* driven.
- ✓ After the interrupt is serviced, the saved return address is loaded into the program counter, and the interrupted computation resumes as though the interrupt is not occurred.



2. Storage Structure:

- ✓ **Main Memory**:

Computer programs must be in main memory (also called **random-access memory** or **RAM)** to be executed. Main memory is the only large storage area (millions to billions of bytes) that the processor can access directly. It commonly is implemented in a semiconductor technology called Dynamic Random Access Memory (DRAM). Other memory devices are ROM, EEPROM etc. Load and store instructions are used to read and write data/instructions between memory & processor.

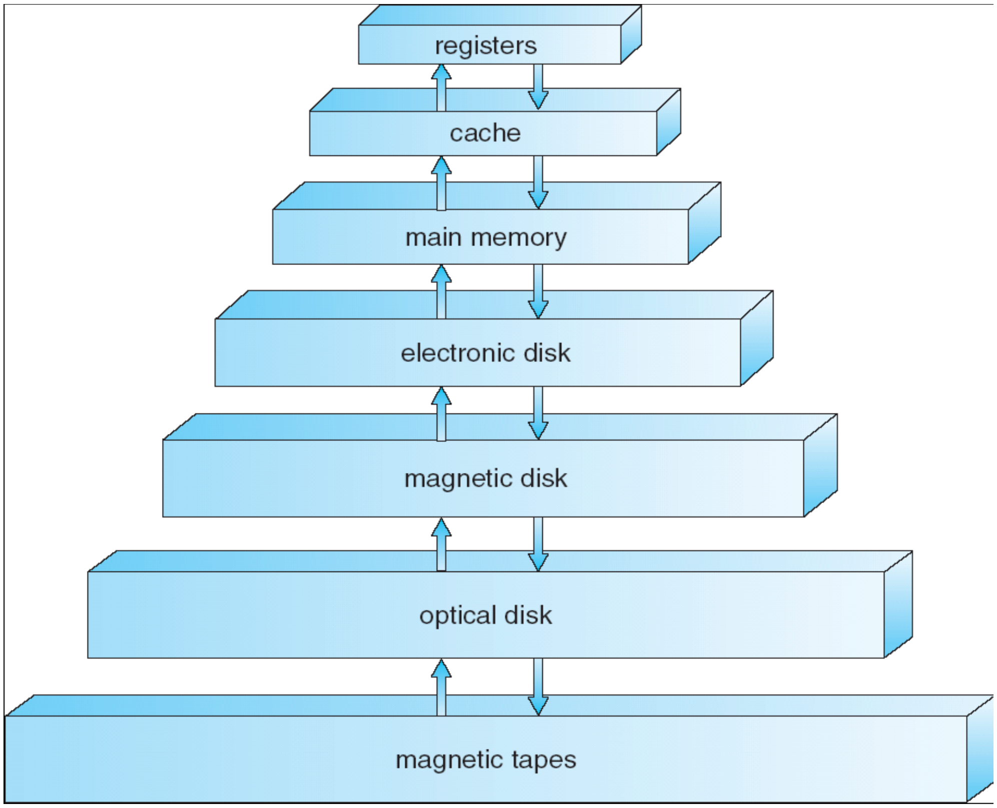
Secondary storage

- ✓ Extension of main memory that provides large nonvolatile storage capacity.

- **Magnetic disks**
  - ✓ Rigid metal or glass platters covered with magnetic recording material
  - ✓ Disk surface is logically divided into *tracks*, which are subdivided into *sectors*.
  - ✓ The *disk controller* determines the logical interaction between the device and the computer.

Other storages include cache, CD-ROM, magnetic tapes and so on. The main differences among the storage systems lie in speed, cost, size and volatility. The storage system can be organized in a hierarchy according to speed and cost. Higher levels are expensive but they are fast. Down in the hierarchy cost decreases and the access time increases.

Explain storage device hierarchy, with neat diagram.



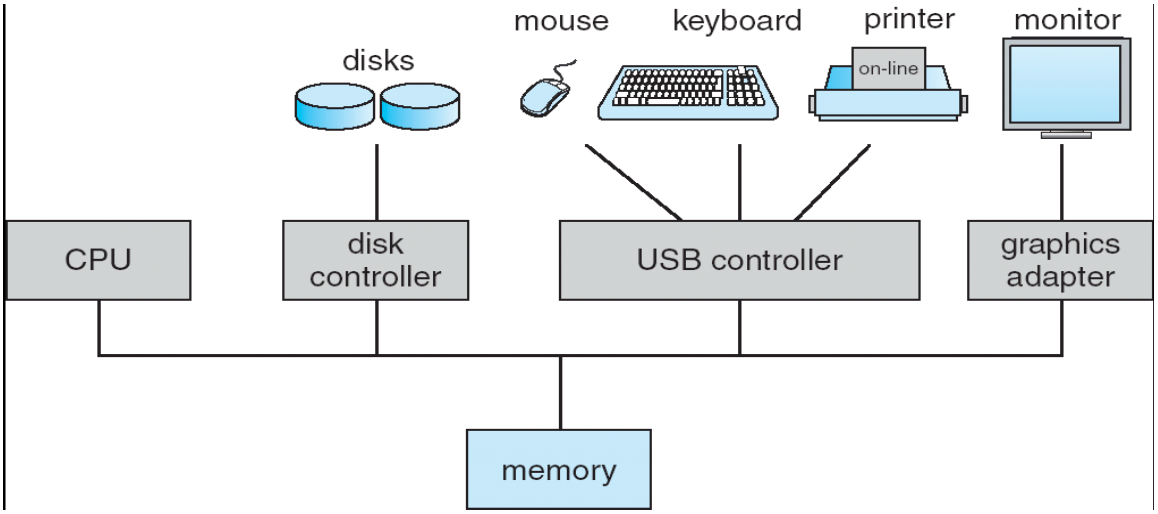
The wide variety of storage systems in a computer system can be organized in a hierarchy as shown above, is according to speed and cost. The higher levels are expensive, but they are fast.

As we move down the hierarchy, the cost per bit generally decreases, whereas the access time generally increases.

CACHING

- Information in use copied from slower to faster storage device temporarily
- Faster storage (cache) checked first to determine if information is there in main memory
  - If it is, information used directly from the cache (fast)
  - If not, data copied to cache and used there.

I/O STRUCTURE



Storage is only one of many types of I/O devices within a computer. A large portion of operating system code is dedicated to managing I/O, both because of its importance to the reliability and performance of a system and because of the varying nature of the devices. A general-purpose computer system consists of CPUs and multiple device controllers that are connected through a common bus. Each device controller is in charge of a specific type of device. Depending on the controller, there may be more than one attached device. The device controller is responsible for moving the data between the peripheral devices that it controls and its local buffer storage. Typically, operating systems have a **device driver** for each device controller. This device driver understands the device controller and presents a uniform interface to the device to the rest of the operating system.

COMPUTER-SYSTEM ARCHITECTURE

A computer system may be organized in a number of different ways, which we can categorize roughly according to the number of general-purpose processors used.

- Single processor systems
- Multiprocessor system
- Clustered system

SINGLE-PROCESSOR SYSTEM

- One main CPU capable of executing a general-purpose instruction set, including instructions from user processes.
- All systems have special purpose processors. i.e device specific processors such as disk, keyboard, and graphics controller
- Mainframes- general purpose processors such as I/O processors
- Run limited set of instruction does not run user processes.
- Use of special purpose processor does not turn a single processor system into multiprocessor system

MULTI-PROCESSOR SYSTEMS

- Also known as parallel systems or tightly coupled systems
- Two or more processors in close communication sharing the computer bus and sometimes the clock, memory and peripheral devices

*****List the three main advantages of multi-processor systems**

Advantages

1. **Increased throughput**: By increasing the number of processors, we expect to get more work done in less time
2. **Economy of scale:** Multiprocessor systems can cost less than equivalent multiple single-processor systems.
3. **Increased reliability:** If functions can be distributed properly among several processors, then the failure of one processor will not halt the system, only slow it down.

Define the following terms

i. **Graceful degradation**

ii. **Fault tolerant systems**

Graceful degradation:

The ability of computer system to continue providing service proportional to the level of surviving hardware is called graceful degradation.

It is the ability of a computer, to maintain limited functionality even when a large portion of its component has been destroyed or rendered inoperative.

Fault tolerance in computer system:

Fault tolerance is the way in which an operating system (OS) responds to a hardware or software failure. The term essentially refers to a system's ability to allow for failures or malfunctions, and this ability may be provided by software, hardware or a combination of both.

Explain the types of multiprocessor system

Two types of Multi-processor systems

i. Asymmetric multi-processing system

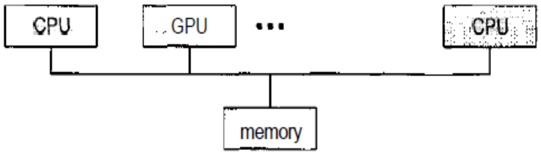
ii. Symmetric multi-processing system

Asymmetric multi-processing system:

- ✓ In this system each processor is assigned a specific task.
- ✓ A master processor controls the system; and other processors either look to the master for instruction or have predefined tasks
- ✓ It defines master-slave relationship

Symmetric multi-processing system:

- ✓ In this system each processor performs all tasks within the operating system.
- ✓ No master-slave relationship exists between processors.



CLUSTERED SYSTEMS

****Explain the types of clustered system

- ✓ Clustered system gather together multiple CPUs to accomplish computational work
- ✓ Clustered systems are composed of two or more individual systems or nodes joined together.
- ✓ Clustered system share storage and are closely linked via local area network.
- ✓ Clustering provide high availability.

Types of clustering systems

i. Asymmetric clustering
ii. Symmetric clustering
iii. Parallel clustering
iv. Clustering over wide area network

Asymmetric clustering

- ✓ One machine is in hot-standby mode while other is running the applications.
- ✓ The hot standby host machine does nothing but monitor the active server.
- ✓ If server fails, the hot standby host machine becomes the active server.

Symmetric clustering

- ✓ Two or more hosts are running applications and are monitoring each other.
- ✓ This mode of system is more efficient, as it uses all of the available hardware.

Parallel clustering

- ✓ Parallel clusters allow multiple hosts to access the shared data.

Clustering over wide area network

- ✓ It is a collection of physically separate computer systems that are networked (WAN) to provide the users with access to the various resources that the system maintains.

- ✓ Access to shared resource increases computation speed, functionality, data availability and reliability.

OPERATING-SYSTEM STRUCTURE

The various types of general purpose OS is

1. Batch Systems

1. Multiprogramming

2. Time sharing or Multitasking

Define the essential features of Batch operating system (Batch file system)

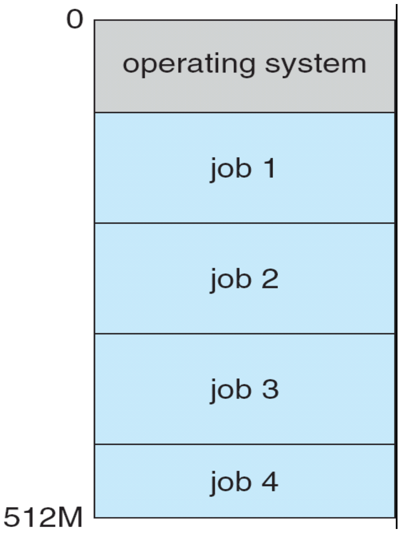
BATCH SYSTEM

- ✓ The user of batch OS do not interact with the computer directly.
- ✓ Each user prepares his job on an off-line device like punch cards and submits it to the computer operator
- ✓ At some later time (after minutes, hours or days) the output is generated.
- ✓ To speed up processing, jobs with similar needs are batched together and run as a group
- ✓ As several jobs are given to computer system and unless it completes the first job it is not possible to process the next job. Therefore it is an in-efficient OS. Also the CPU is often idle because the speed of the mechanical I/O devices.

Define the essential features of Multiprogramming operating system

MULTIPROGRAMMING

- ✓ **Multiprogramming** increases CPU utilization by organizing jobs (code and data) so that the CPU always has one to execute.
- ✓ OS keeps several jobs in memory simultaneously.
- ✓ One job selected and run via **job scheduling**
- ✓ In a multi-programmed system, the operating system simply switches to, and executes, another job. When *that* job needs to wait, the CPU is switched to *another* job, and so on.
- ✓ As long as at least one job needs to execute, the CPU is never idle.
- ✓ Multiprogramming provides an environment in which various system resources are utilized effectively but they do not provide user interaction with the computer system.



TIME SHARING OR MULTITASKING

Define the essential features of Time sharing operating system

- ✓ In time-sharing systems, the CPU executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while it is running.
- ✓ The user gives instructions to the operating system or to a program directly, using a input device such as a keyboard or a mouse, and waits for immediate results on an output device. Accordingly, the **response time** should be short—typically less than one second.
- ✓ A time-shared operating system allows many users to share the computer simultaneously.
- ✓ A time-shared operating system uses CPU scheduling.

****Differentiate between Multi-programming, Multi-tasking (Time sharing) and Multi-processing system

S. No	Multiprogramming	Multitasking	Multiprocessing
1	Multiprogramming is the ability of an operating system to execute more than one program on a single processor machine based on context switching.	Multitasking is the ability of an operating system to execute more than one task simultaneously on a single processor machine based on time sharing.	Multiprocessing is the ability of an operating system to execute more than one process simultaneously on a multi processor machine by having more than one physical processor.
2	Multiple programs reside in the main memory simultaneously to improve CPU utilization so that CPU does not sit idle for a long time.	It enables execution of multiple tasks and processes at the same time to increase CPU performance.	It utilizes multiple CPUs for process allocation
3	The main idea is to keep the CPU busy as long as there are processes ready to execute.	The idea is to allow multiple processes to run simultaneously via time sharing	The idea is to allow multiple processes to run simultaneously via multiple processors.
4	They do not provide user interaction with the computer system	It allows many users to share the computer simultaneously	It provides user interaction with the system

OPERATING SYSTEM OPERATIONS

Modern Operating system is interrupt driven. Events are always signaled by the occurrence of an interrupt or trap. A trap is a software generated interrupt caused either by error (Division by zero, invalid memory access) or request from a user program that an OS service be performed. For each type of interrupt, separate segments of code in the OS determine what action should be taken. The Interrupt Service Routine (ISR) is provided that is responsible for dealing with interrupt.

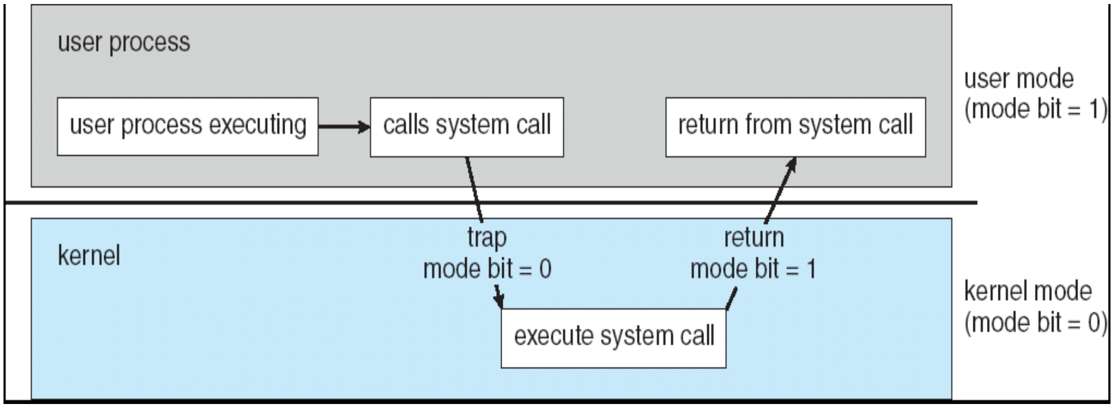
DUAL MODE OPERATION OF OPERATING SYSTEM

*****Explain the operation of operating system

*****With neat diagram explain the Dual mode operation of operating system

In order to differentiate between the execution of OS code and user defined code, most of the computers provide hardware support by executing these codes in two separate modes

1. User Mode
2. Kernel or Supervisor or Privileged or System mode



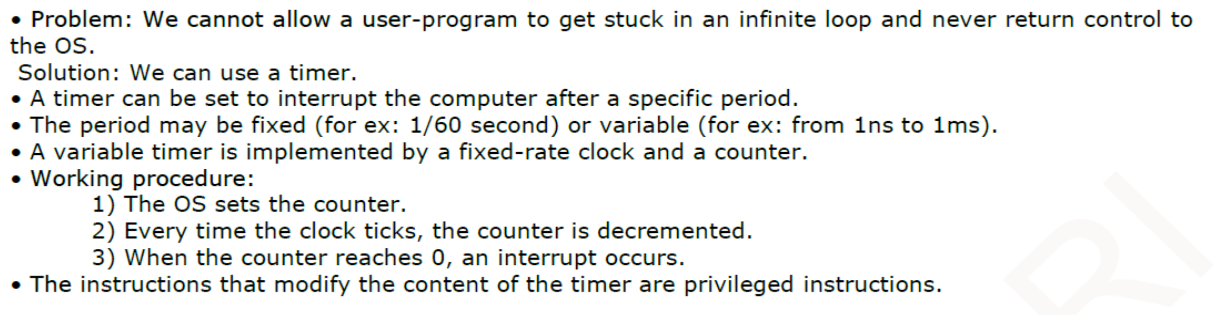
A mode bit is added to the computer hardware to indicate the current mode; ie: mode bit = 0 for kernel mode and 1 for user mode.

Working Principle:

- ✓ At system Boot time, the hardware starts in kernel mode
- ✓ The OS is then loaded and starts user application in user mode
- ✓ Whenever a trap or interrupt occurs, the hardware switches from user mode to kernel mode

- ✓ The system always switches to user mode (setting mode bit = 1), before passing control to user program
- ✓ Dual mode operation protects OS from errant users and errant user from one another
- ✓ Privileged instruction is executed only in kernel mode
- ✓ If an attempt is made to execute privileged instruction in user mode, the hardware treats it as illegal and traps it to the OS
- ✓ A system calls are called by user program to ask the OS to perform the tasks on behalf of the user program.

TIMER



- Problem: We cannot allow a user-program to get stuck in an infinite loop and never return control to the OS.

  Solution: We can use a timer.
- A timer can be set to interrupt the computer after a specific period.
- The period may be fixed (for ex: 1/60 second) or variable (for ex: from 1ns to 1ms).
- A variable timer is implemented by a fixed-rate clock and a counter.
- Working procedure:
  1) The OS sets the counter.
  2) Every time the clock ticks, the counter is decremented.
  3) When the counter reaches 0, an interrupt occurs.
- The instructions that modify the content of the timer are privileged instructions.

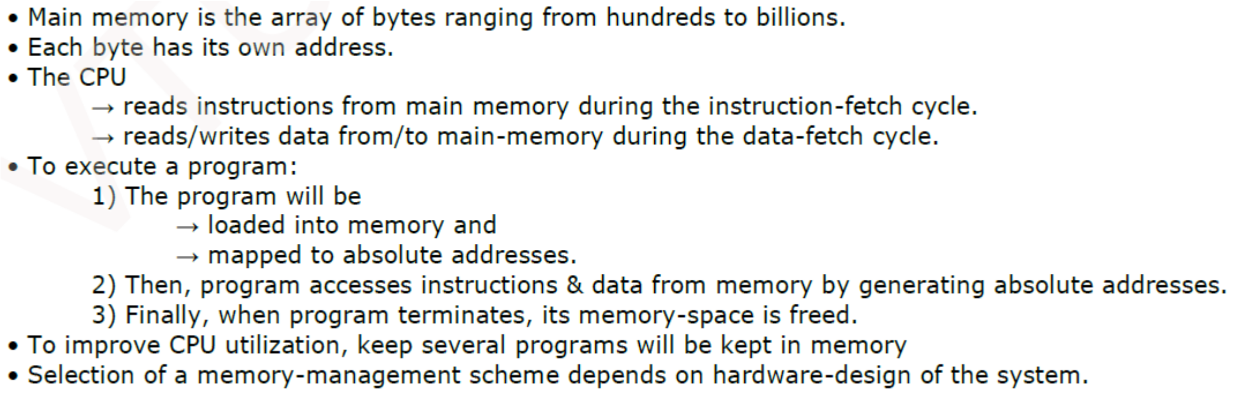
PROCESS MANAGEMENT

A process is a program in execution. It is a unit of work within the system. Program is a *passive entity*, process is an *active entity*. Process needs resources to accomplish its task like CPU, memory, I/O, files. Process termination requires reclaim of any reusable resources. Single-threaded process has one program counter specifying location of next instruction to execute. Process executes instructions sequentially, one at a time, until completion. Multi-threaded process has one program counter per thread. Typically system has many processes, some user, some operating system running concurrently on one or more CPUs. All processes can execute concurrently by multiplexing the single CPUs.

*****What are the operating System activities connected with Process management

1. Creating and deleting both user and system processes
2. Suspending and resuming processes
3. Providing mechanisms for process synchronization
4. Providing mechanisms for process communication
5. Providing mechanisms for deadlock handling

MEMORY MANAGEMENT



- Main memory is the array of bytes ranging from hundreds to billions.
- Each byte has its own address.
- The CPU
  - → reads instructions from main memory during the instruction-fetch cycle.
  - → reads/writes data from/to main-memory during the data-fetch cycle.
- To execute a program:
  1) The program will be
     - → loaded into memory and
     - → mapped to absolute addresses.
  2) Then, program accesses instructions & data from memory by generating absolute addresses.
  3) Finally, when program terminates, its memory-space is freed.
- To improve CPU utilization, keep several programs will be kept in memory
- Selection of a memory-management scheme depends on hardware-design of the system.

*****What are the operating System activities connected with Memory management

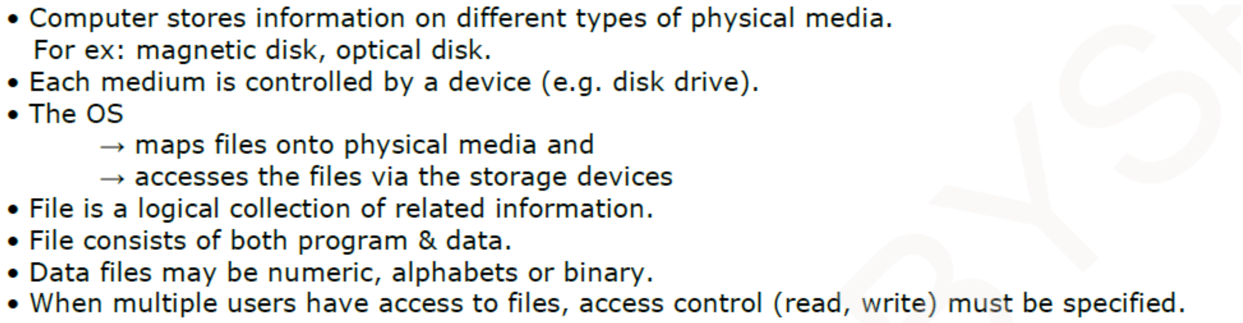
Memory management activities

1. Keeping track of which parts of memory are currently being used and by whom
2. Deciding which processes (or parts thereof) and data to move into and out of memory
3. Allocating and de-allocating memory space as needed

STORAGE MANAGEMENT

1. Storage management
2. Mass-Storage management
3. Caching

FILE MANAGEMENT



- Computer stores information on different types of physical media.
  For ex: magnetic disk, optical disk.
- Each medium is controlled by a device (e.g. disk drive).
- The OS
  - → maps files onto physical media and
  - → accesses the files via the storage devices
- File is a logical collection of related information.
- File consists of both program & data.
- Data files may be numeric, alphabets or binary.
- When multiple users have access to files, access control (read, write) must be specified.

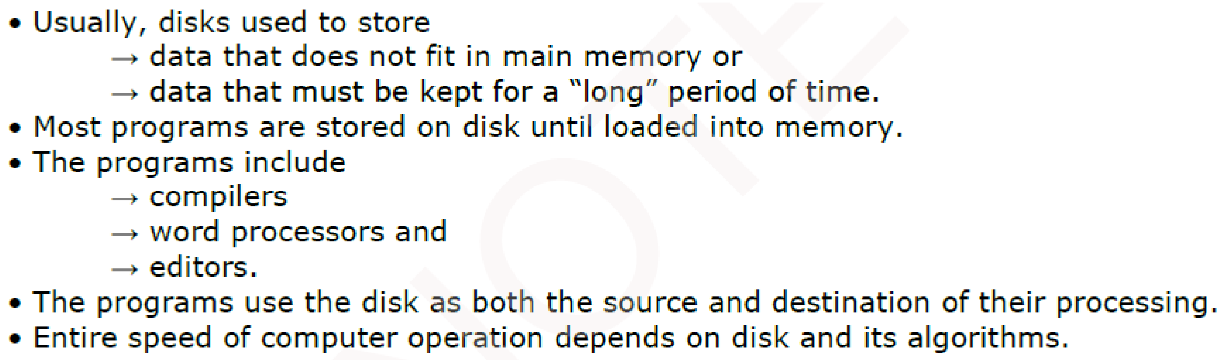
*****What are the operating System activities connected with File management

OS activities include

1. Creating and deleting files and directories
2. Supporting Primitives to manipulate files and directories

3. Mapping files onto secondary storage
4. Backup files onto stable (non-volatile) storage media

MASS-STORAGE MANAGEMENT



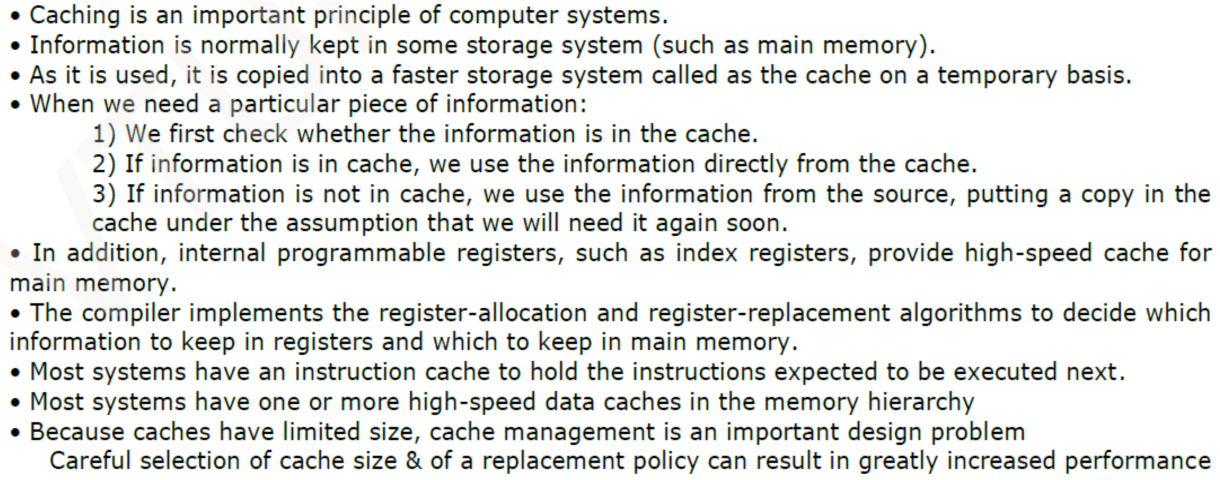
- Usually, disks used to store
  → data that does not fit in main memory or
  → data that must be kept for a "long" period of time.
- Most programs are stored on disk until loaded into memory.
- The programs include
  → compilers
  → word processors and
  → editors.
- The programs use the disk as both the source and destination of their processing.
- Entire speed of computer operation depends on disk and its algorithms.

*****What are the operating System activities connected with Mass storage management

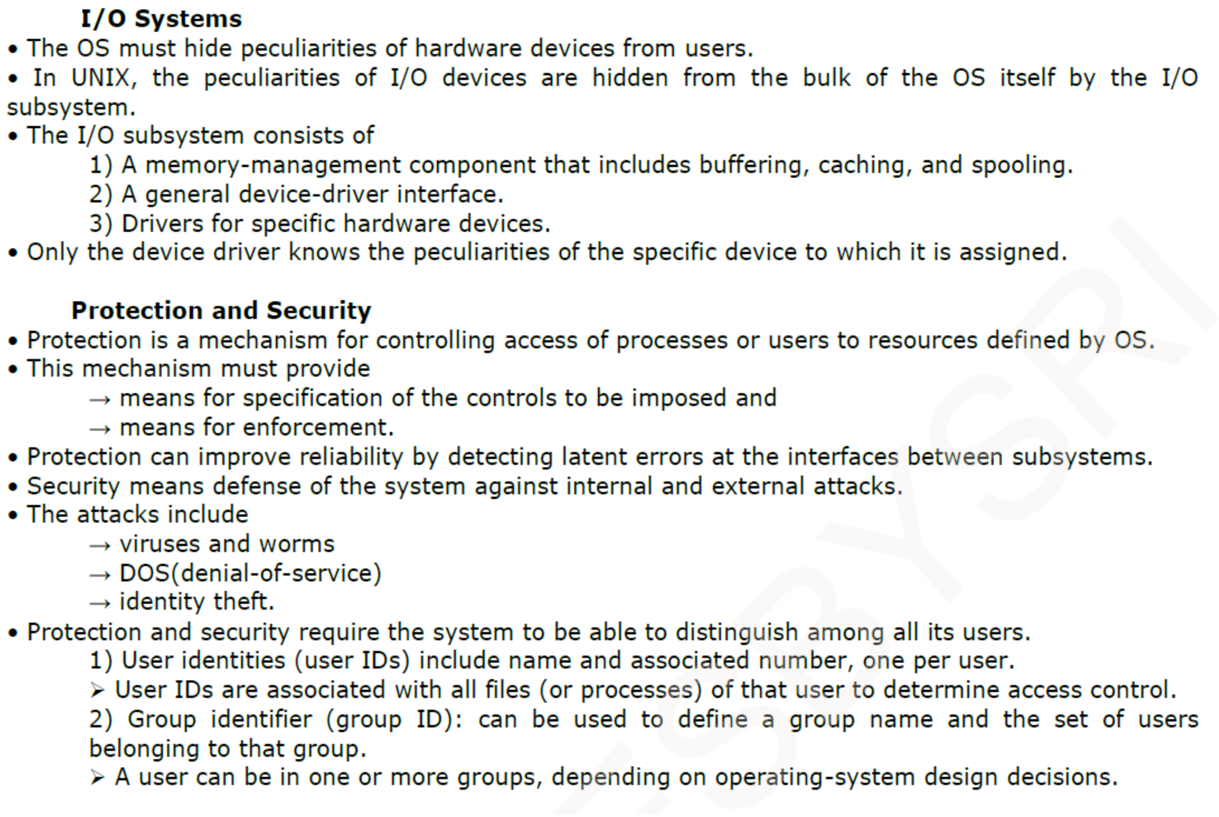
OS activities

1. Free-space management
2. Storage allocation
3. Disk scheduling

CACHING



- Caching is an important principle of computer systems.
- Information is normally kept in some storage system (such as main memory).
- As it is used, it is copied into a faster storage system called as the cache on a temporary basis.
- When we need a particular piece of information:
  1) We first check whether the information is in the cache.
  2) If information is in cache, we use the information directly from the cache.
  3) If information is not in cache, we use the information from the source, putting a copy in the cache under the assumption that we will need it again soon.
- In addition, internal programmable registers, such as index registers, provide high-speed cache for main memory.
- The compiler implements the register-allocation and register-replacement algorithms to decide which information to keep in registers and which to keep in main memory.
- Most systems have an instruction cache to hold the instructions expected to be executed next.
- Most systems have one or more high-speed data caches in the memory hierarchy
- Because caches have limited size, cache management is an important design problem
  Careful selection of cache size & of a replacement policy can result in greatly increased performance



I/O Systems

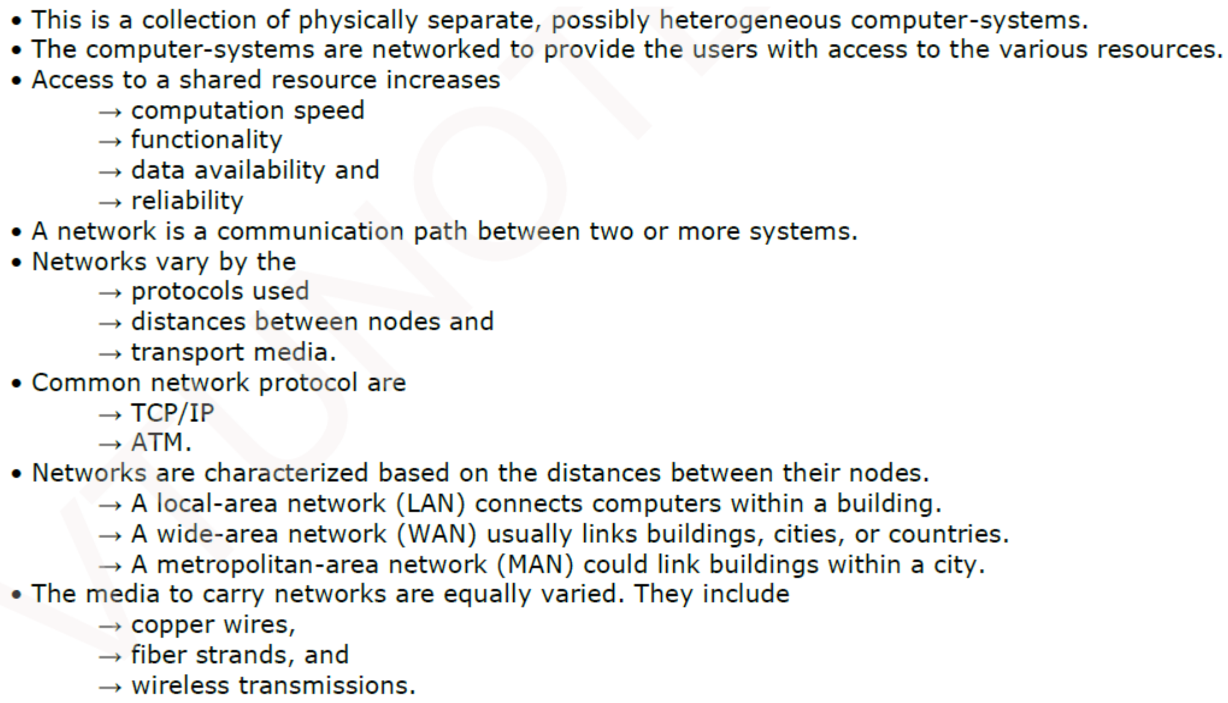
- The OS must hide peculiarities of hardware devices from users.
- In UNIX, the peculiarities of I/O devices are hidden from the bulk of the OS itself by the I/O subsystem.
- The I/O subsystem consists of
  1) A memory-management component that includes buffering, caching, and spooling.
  2) A general device-driver interface.
  3) Drivers for specific hardware devices.
- Only the device driver knows the peculiarities of the specific device to which it is assigned.

Protection and Security

- Protection is a mechanism for controlling access of processes or users to resources defined by OS.
- This mechanism must provide
  → means for specification of the controls to be imposed and
  → means for enforcement.
- Protection can improve reliability by detecting latent errors at the interfaces between subsystems.
- Security means defense of the system against internal and external attacks.
- The attacks include
  → viruses and worms
  → DOS(denial-of-service)
  → identity theft.
- Protection and security require the system to be able to distinguish among all its users.
  1) User identities (user IDs) include name and associated number, one per user.
  ➢ User IDs are associated with all files (or processes) of that user to determine access control.
  2) Group identifier (group ID): can be used to define a group name and the set of users belonging to that group.
  ➢ A user can be in one or more groups, depending on operating-system design decisions.

DISTRIBUTED SYSTEM

**List the essential properties of Distributed OS



- This is a collection of physically separate, possibly heterogeneous computer-systems.
- The computer-systems are networked to provide the users with access to the various resources.
- Access to a shared resource increases
  → computation speed
  → functionality
  → data availability and
  → reliability
- A network is a communication path between two or more systems.
- Networks vary by the
  → protocols used
  → distances between nodes and
  → transport media.
- Common network protocol are
  → TCP/IP
  → ATM.
- Networks are characterized based on the distances between their nodes.
  → A local-area network (LAN) connects computers within a building.
  → A wide-area network (WAN) usually links buildings, cities, or countries.
  → A metropolitan-area network (MAN) could link buildings within a city.
- The media to carry networks are equally varied. They include
  → copper wires,
  → fiber strands, and
  → wireless transmissions.

SPECIAL PURPOSE OPERATING SYSTEMS

These are the systems whose functions are limited and whose objective is to deal with limited computation.

1. Real-time Embedded System
2. Multimedia systems
3. Handheld systems

REAL-TIME EMBEDDED SYSTEM (Real Time OS)

***List the essential features of Real time OS

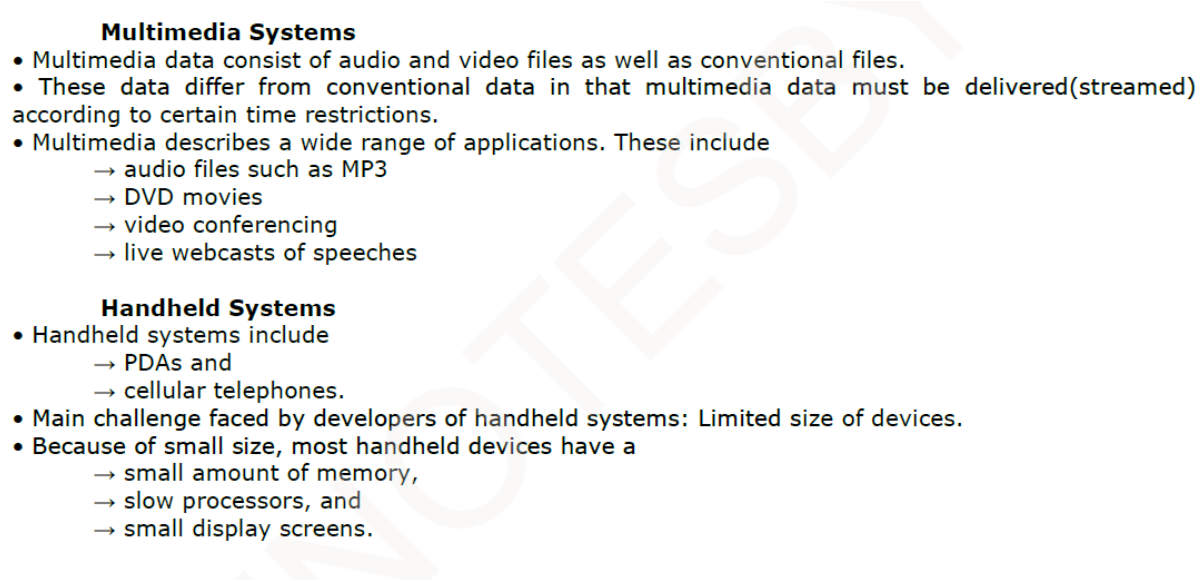
- ✓ RTOS tend to have specific task(monitoring and managing other devices)
- ✓ RTOS provide little or no user interface
- ✓ Real time OS has well-defined fixed time constraints. Processing must be done within the defined constraints, or the system will fail.
- ✓ It is often used as control device in dedicated application.

Embedded systems are most prevalent form of computers

Embedded system varies as

- General purpose system running standard OS
- Hardware devices with special purpose OS

The use of embedded devices continues to expand. Embedded system always run on real-time operating systems.

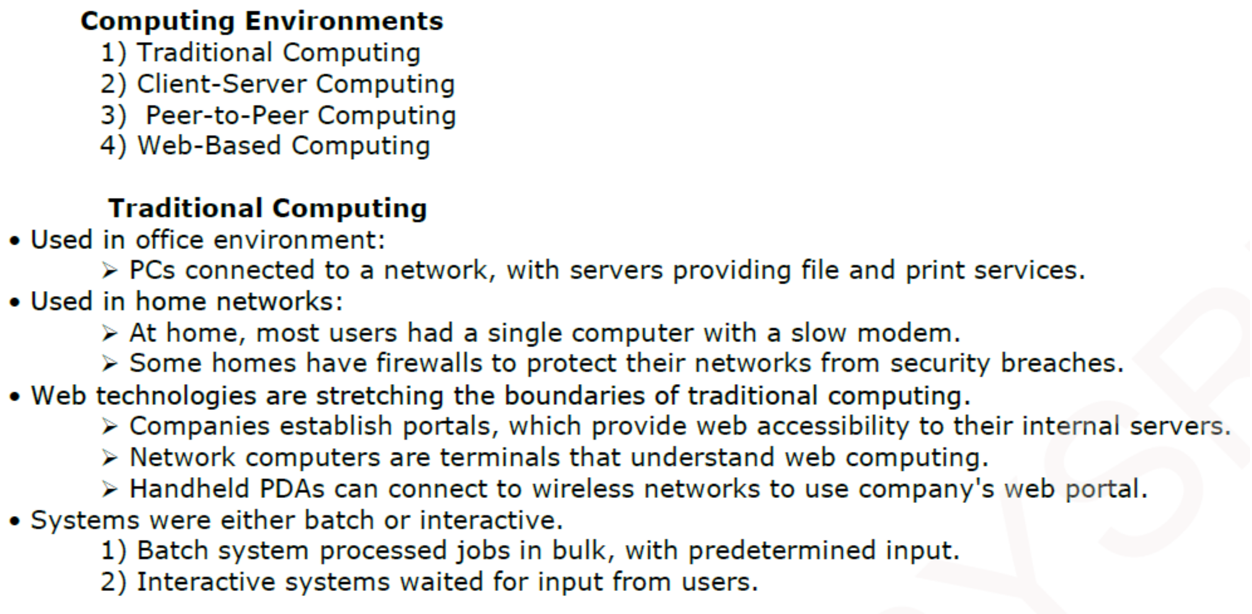


Multimedia Systems

- Multimedia data consist of audio and video files as well as conventional files.
- These data differ from conventional data in that multimedia data must be delivered(streamed) according to certain time restrictions.
- Multimedia describes a wide range of applications. These include
  → audio files such as MP3
  → DVD movies
  → video conferencing
  → live webcasts of speeches

Handheld Systems

- Handheld systems include
  → PDAs and
  → cellular telephones.
- Main challenge faced by developers of handheld systems: Limited size of devices.
- Because of small size, most handheld devices have a
  → small amount of memory,
  → slow processors, and
  → small display screens.



Computing Environments

1) Traditional Computing
2) Client-Server Computing
3) Peer-to-Peer Computing
4) Web-Based Computing

Traditional Computing

- Used in office environment:
  ➢ PCs connected to a network, with servers providing file and print services.
- Used in home networks:
  ➢ At home, most users had a single computer with a slow modem.
  ➢ Some homes have firewalls to protect their networks from security breaches.
- Web technologies are stretching the boundaries of traditional computing.
  ➢ Companies establish portals, which provide web accessibility to their internal servers.
  ➢ Network computers are terminals that understand web computing.
  ➢ Handheld PDAs can connect to wireless networks to use company's web portal.
- Systems were either batch or interactive.
  1) Batch system processed jobs in bulk, with predetermined input.
  2) Interactive systems waited for input from users.

Client-Server Computing

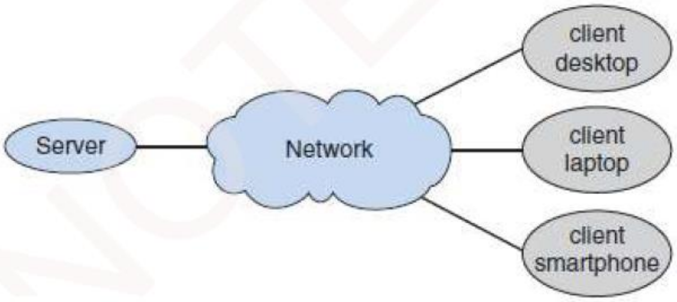
- Servers can be broadly categorized as (Figure 1.10): 1) Compute servers and
  2) File servers

1) Compute-server system provides an interface to which a client can send a request to perform an action (for example, read data).

- ➢ In response, the server executes the action and sends back results to the client.

2) File-server system provides a file-system interface where clients can create, read, and delete files.

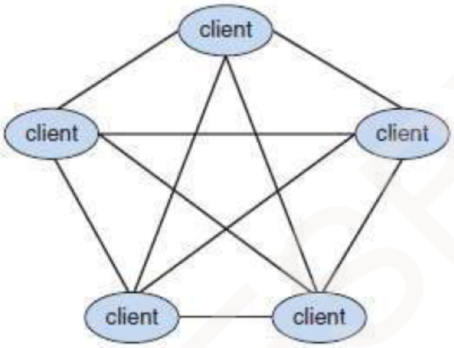
- ➢ For example: web server that delivers files to clients running web browsers.



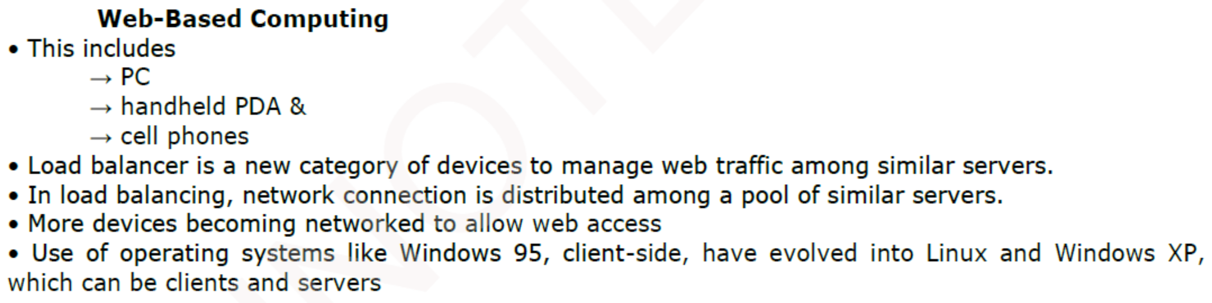
General structure of a client–server system.

Peer-to-Peer Computing

- All nodes are considered peers, and each may act as either a client or a server(Figure 1.11).
- Advantage:
  1) In a client-server system, the server is a bottleneck;
  but in a peer-to-peer system, services can be provided by several nodes distributed throughout the network.
- A node must first join the network of peers.
- Determining what services are available is done in one of two general ways:
  1) When a node joins a network, it registers its service with a centralized lookup service on the network.
  ➢ Any node desiring a specific service first contacts this centralized lookup service to determine which node provides the service.
  2) A peer broadcasts a request for the service to all other nodes in the network. The node (or nodes) providing that service responds to the peer.



Peer-to-peer system with no centralized service.



Web-Based Computing

- This includes
  - → PC
  - → handheld PDA &
  - → cell phones
- Load balancer is a new category of devices to manage web traffic among similar servers.
- In load balancing, network connection is distributed among a pool of similar servers.
- More devices becoming networked to allow web access
- Use of operating systems like Windows 95, client-side, have evolved into Linux and Windows XP, which can be clients and servers

OPERATING-SYSTEM SERVICES

An operating system provides an environment for the execution of programs. It provides certain services to programs and to the users of those programs.

***********List and explain the services provided by OS for the user and efficient operation of system.**

Services provided by the OS that are helpful to *the* user:

1. **User Interface**: Almost all operating systems have a user interface (UI), such as Command-Line (CLI) uses text commands, Graphics User Interface (GUI) is window system, Batch Interface uses commands and directives.
2. **Program execution:** The system must be able to load a program into memory and to run that program, end execution, either normally or abnormally (indicating error)
3. **I/O operations:** A running program may require I/O, which may involve a file or an I/O device.

4. **File-system manipulation:** The file system is of particular interest. Obviously, programs need to read and write files and directories, create and delete them, search them, list file Information, permission management.
5. **Communications**: Processes may exchange information, on the same computer or between computers over a network. Communications may be via shared memory or through message passing (packets moved by the OS)
6. **Error detection** – OS needs to be constantly aware of possible errors. Errors may occur in CPU and memory, in I/O devices and in user program.

Services provided by the OS that are helpful to *the efficient operation of system.*

1. **Resource allocation -** When multiple users or multiple jobs running concurrently, resources must be allocated to each of them
2. **Accounting -** To keep track of which users use how much and what kinds of computer resources
3. **Protection and security** – When several separate processes execute concurrently, it should not be possible for one process to interfere with the others or with the OS itself. **Protection** involves ensuring that all access to system resources is controlled. **Security** of the system from outsiders requires user authentication, extends to defending external I/O devices from invalid access attempts

COMMAND-LINE INTERFACE OR COMMAND INTERPRETER

CLI allows direct command entry

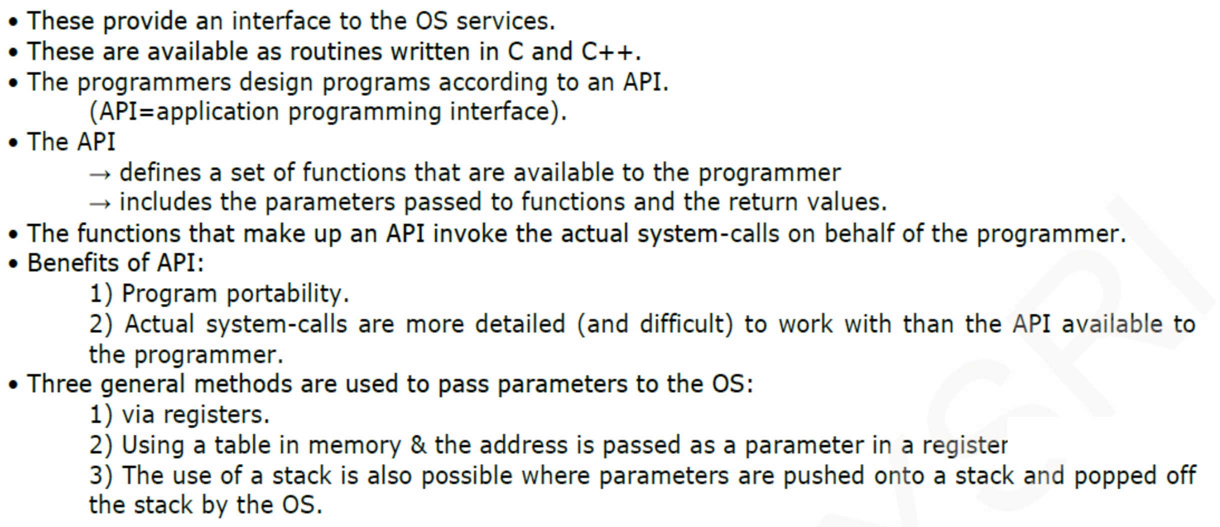
- Sometimes implemented in kernel, sometimes by systems program
- Sometimes multiple flavors implemented – **shells**
- Primarily fetches a command from user and executes it
- Sometimes commands built-in, sometimes just names of programs

GRAPHICAL USER INTERFACE

- User-friendly **desktop** metaphor interface
  - Usually mouse, keyboard, and monitor
  - **Icons** represent files, programs, actions, etc
  - Various mouse buttons over objects in the interface cause various actions (provide information, options, execute function, open directory (known as a **folder**)

- Many systems now include both CLI and GUI interfaces
  - Microsoft Windows is GUI with CLI “command” shell
  - Solaris is CLI with optional GUI interfaces (Java Desktop, KDE)

SYSTEM CALLS



- These provide an interface to the OS services.
- These are available as routines written in C and C++.
- The programmers design programs according to an API.
  (API=application programming interface).
- The API
  → defines a set of functions that are available to the programmer
  → includes the parameters passed to functions and the return values.
- The functions that make up an API invoke the actual system-calls on behalf of the programmer.
- Benefits of API:
  1) Program portability.
  2) Actual system-calls are more detailed (and difficult) to work with than the API available to the programmer.
- Three general methods are used to pass parameters to the OS:
  1) via registers.
  2) Using a table in memory & the address is passed as a parameter in a register
  3) The use of a stack is also possible where parameters are pushed onto a stack and popped off the stack by the OS.

*****What are System calls? Explain the types of system call.

System calls provides an interface between a process and operating system to allow user-level processes to request services of the operating system.

There are 5 different types of system calls:

1. Process control System call
2. File management
3. Device management
4. Information maintenance
5. Communication.

Process control System calls are used:

- ✓ End , abort
- ✓ Load, Execute
- ✓ Create process, terminate process
- ✓ get process attributes, set process attributes

- ✓ Wait for time
- ✓ Wait event, signal event
- ✓ Acquire lock and release lock

Common system calls dealing with file management are

- ✓ Create file, delete file
- ✓ Open, close
- ✓ Read, write, reposition
- ✓ get file attributes(file name, file type, protection codes, accounting information), set file attributes
- ✓ Move and copy(API, system programs)

Common system calls dealing with Device management are

- ✓ Request device, release device
- ✓ Read, write, reposition
- ✓ Get device attributes, set device attributes
- ✓ Logically attach or detach devices

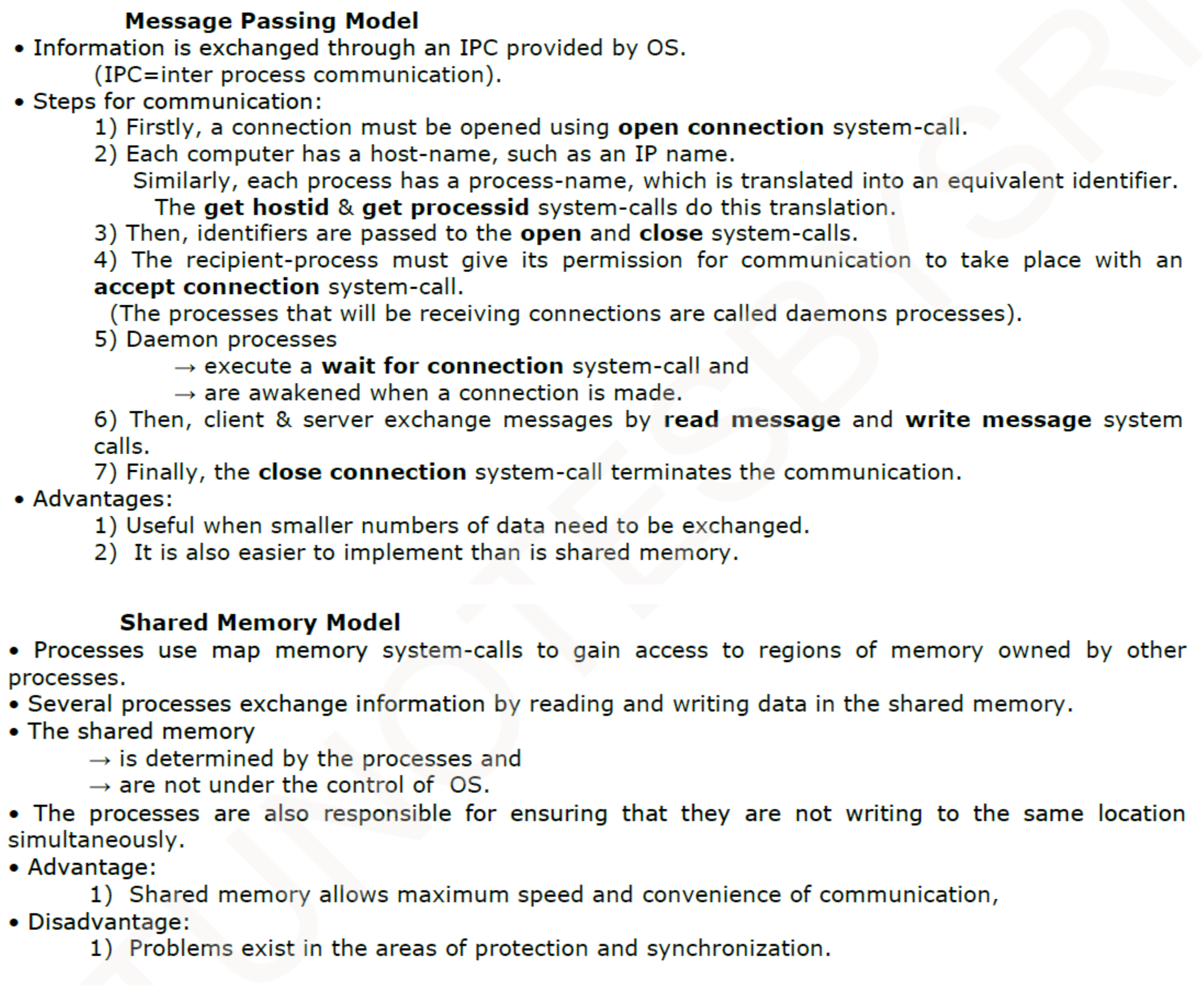
Common system calls dealing with Information maintenance are

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

Common system calls dealing with Communication are

- ✓ Create, delete communication connection
- ✓ Send, receive messages
- ✓ Transfer status information
- ✓ Attach or detach remote devices

Two modes of communication: 1. Message passing model 2. Shared memory model



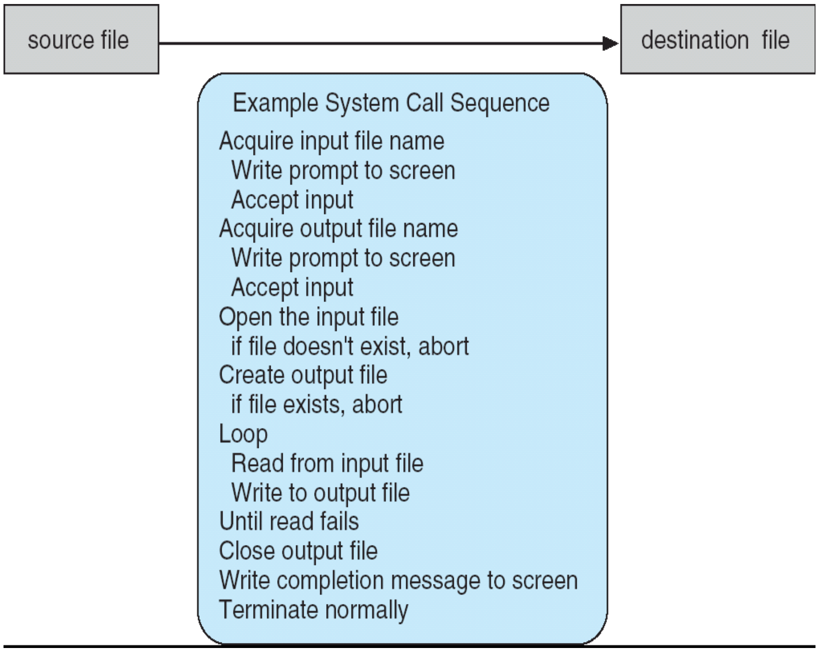
Message Passing Model

- Information is exchanged through an IPC provided by OS.
  (IPC=inter process communication).
- Steps for communication:
  1) Firstly, a connection must be opened using **open connection** system-call.
  2) Each computer has a host-name, such as an IP name.
     Similarly, each process has a process-name, which is translated into an equivalent identifier.
     The **get hostid** & **get processid** system-calls do this translation.
  3) Then, identifiers are passed to the **open** and **close** system-calls.
  4) The recipient-process must give its permission for communication to take place with an **accept connection** system-call.
     (The processes that will be receiving connections are called daemons processes).
  5) Daemon processes
     → execute a **wait for connection** system-call and
     → are awakened when a connection is made.
  6) Then, client & server exchange messages by **read message** and **write message** system calls.
  7) Finally, the **close connection** system-call terminates the communication.
- Advantages:
  1) Useful when smaller numbers of data need to be exchanged.
  2) It is also easier to implement than is shared memory.

Shared Memory Model

- Processes use map memory system-calls to gain access to regions of memory owned by other processes.
- Several processes exchange information by reading and writing data in the shared memory.
- The shared memory
  → is determined by the processes and
  → are not under the control of OS.
- The processes are also responsible for ensuring that they are not writing to the same location simultaneously.
- Advantage:
  1) Shared memory allows maximum speed and convenience of communication,
- Disadvantage:
  1) Problems exist in the areas of protection and synchronization.

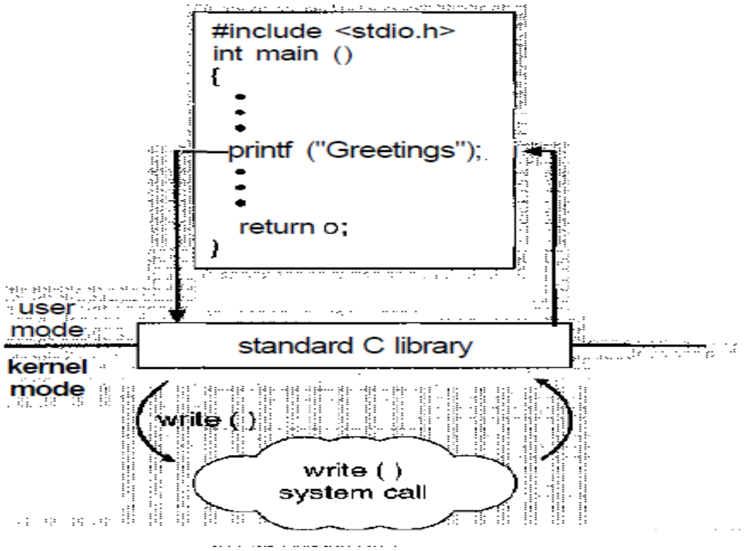
Write the system call sequence to copy a file from source to destination



```
Acquire input file name
  Write prompt to screen
  Accept input
Acquire output file name
  Write prompt to screen
  Accept input
Open the input file
  if file doesn't exist, abort
Create output file
  if file exists, abort
Loop
  Read from input file
  Write to output file
Until read fails
Close output file
Write completion message to screen
Terminate normally
```

With example explain how system call handles user applications

Let us consider an example, a C program invokes the printf () statement. The C library intercepts this call and invokes the necessary system call in the operating system, in this instance, the write () system call. The C library takes the value returned by write () and passes it back to the user program. This is shown in below figure.



```
#include <stdio.h>
int main ()
{
   •
   •
   •
  printf ("Greetings");
   •
   •
   •
   return 0;
}
```

SYSTEM PROGRAM

***What are the different categories of system program

Six categories of system program are:

1. File management: These programs manipulate files. Create, delete, copy, rename, print, dump, list etc
2. Status information: Some programs ask the system for Date(time), amount of memory, number of users etc.
3. File modification: text editors can be used to create and modify the content of files stored on the disk.
4. Programming Language support: Compilers, assemblers and interpreters for common programming languages are provided to the user
5. Program loading and Execution: the system may provide absolute loader, re-locatable loaders, linkage editors, overlay loaders, Debugger etc.
6. Communications: These programs are used for creating virtual connections between processes, users and computer systems. They allow user to browse web pages, send email or log-in remotely

OPERATING SYSTEM DESIGN AND IMPLEMENTATION

Affected by choice of hardware, type of system-batch system, time shared system

User goals and System goals

User goals – operating system should be convenient to use, easy to learn, reliable, safe, and fast

System goals – operating system should be easy to design, implement, and maintain, as well as flexible, reliable, error-free, and efficient. The wide variety of systems in existence shows different requirements can result in variety of solutions for different environments.

- ✓ Specifying and designing an OS is a highly creative task.
- ✓ Software engineering provide general principles.

MECHANISM AND POILCIES

One important principle is the separation of policy and mechanism.

- ✓ **Policy:** What will be done?
- ✓ **Mechanism:** How to do it?

Is separation of mechanism and policy desirable while designing an operating system? Discuss with an example.

The separation of mechanism and policy is important for flexibility. Policies are likely to change across places or time. A general mechanism insensitive to changes in policy would be more desirable. A change in policy would then require redefinition of only certain parameters of the system. Thus mechanisms determine how to do something and policies decide what will be done.

Example: Timer construct.

Mechanism: CPU protection

Policy: How long the timer is to be set for particular user.

- Timer to prevent infinite loop / process hogging resources
- Before turning over control to the user, the operating system ensures that the timer is set to interrupt.
  - ✓ Set interrupt after specific period
  - ✓ Operating system decrements counter
  - ✓ When counter zero generate an interrupt
  - ✓ Set up before scheduling process to regain control or terminate program that exceeds allotted time.

IMPLEMENTATION

Once an OS is designed it must be implemented. Traditionally OS is implemented in assembly language and now it is written in c or c++ .

Example: MS-DOS written in intel 8088 assembly language. Linux OS written in C is available to number of CPUs

OPERATING SYSTEM STRUCTURE

Modern operating system is designed so as to function properly and be modified easily. Common approach is to partition the task into small components rather than having one monolithic system. The various structures of operating systems are:

1. Simple structured OS
2. Layered Approach
3. Micro-kernels
4. Modules

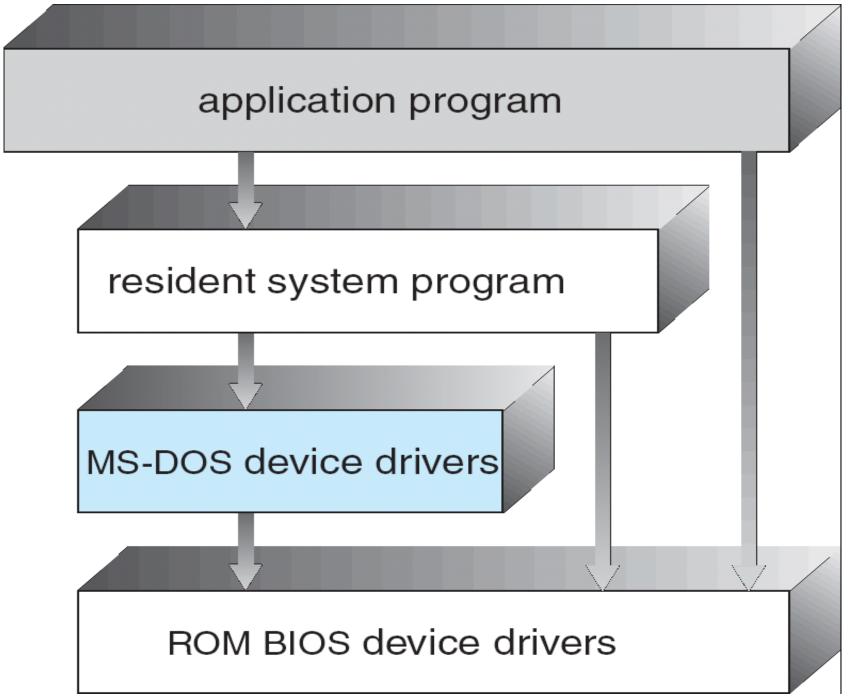
SIMPLE STRUCTURED OS

Explain the simple structured operating system with neat diagram

- ✓ These OS are simple, small and limited systems
- ✓ MS-DOS and UNIX is the examples for simple OS
- ✓ MS-DOS which is written to provide the most functionality in the least space

Disadvantages:

- ✓ It was not divided into modules carefully
- ✓ The interfaces and levels of functionality are not well separated



Explain the simple structured UNIX-operating system with neat diagram

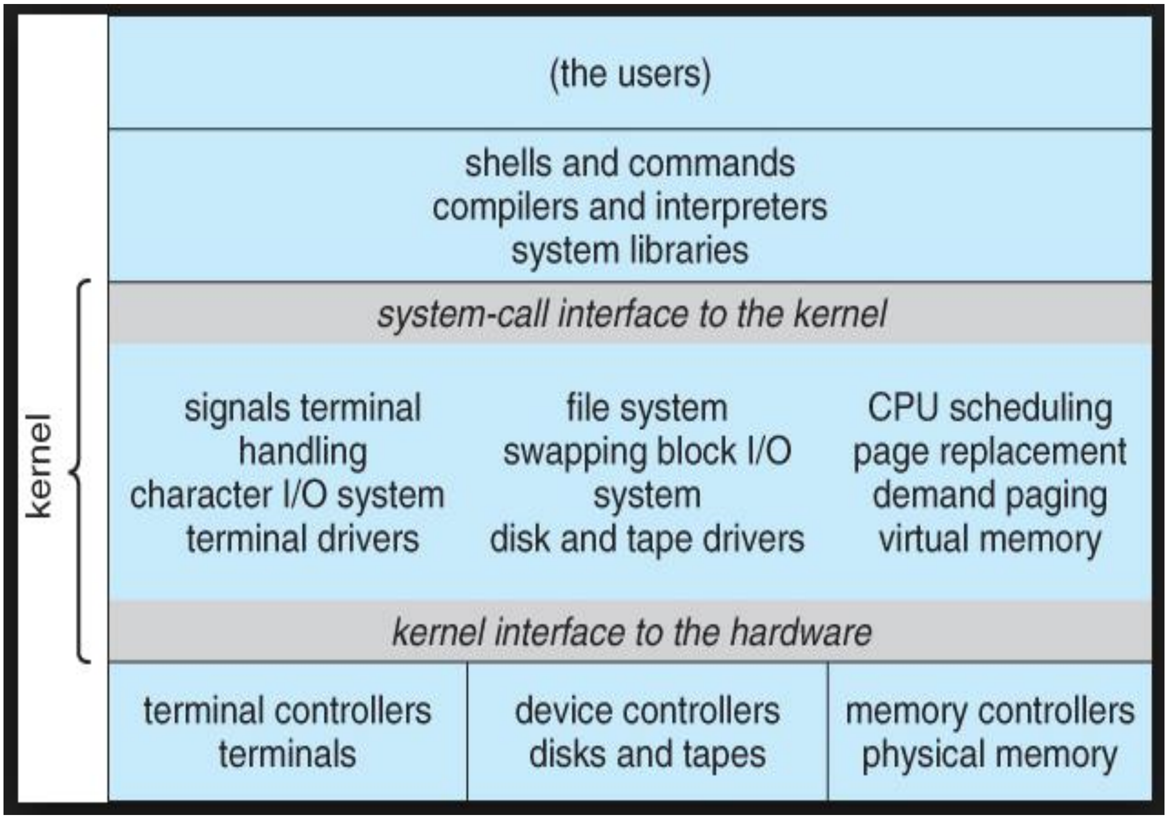
UNIX OS has two parts:

1. Kernel and
2. System programs.

- ✓ The kernel is further separated into a series of interfaces and device drivers.
- ✓ Everything below the system-call interface and above the physical hardware is the kernel.
- ✓ The kernel provides following functions through system calls:
  - → file system
  - → CPU scheduling and
  - → Memory management.

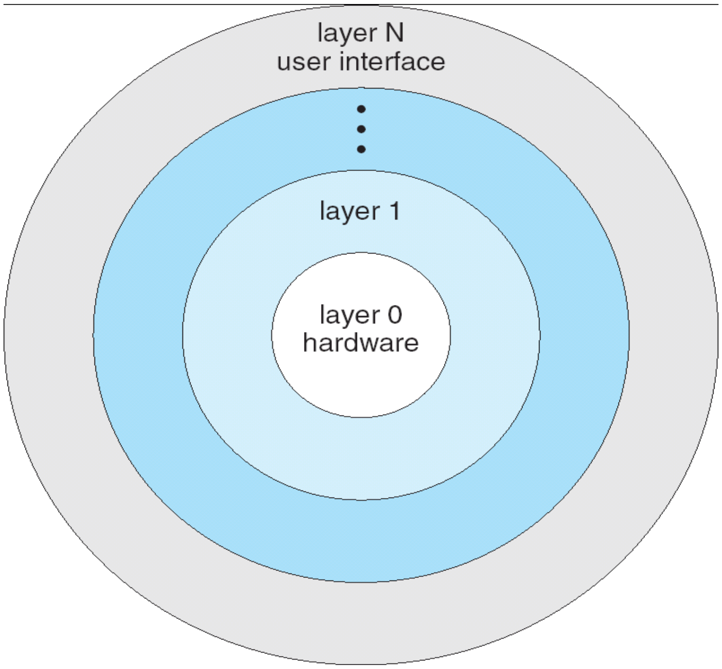
Disadvantage:

- ✓ Difficult to enhance, as changes in one section badly affects other areas.



LAYERED APPROACH

Explain the advantages of layered approach OS with neat diagram



- ✓ The OS is divided into a number of layers.
- ✓ Each layer is built on the top of another layer.
- ✓ The bottom layer is the hardware.
  - The highest is the user interface
- ✓ A layer is an implementation of an abstract-object.
  - i.e. The object is made up of
    - data and
    - operations that can manipulate the data.
- ✓ The layer consists of a set of routines that can be invoked by higher-layers. Higher-layer
  - does not need to know how lower-layer operations are implefmented
  - needs to know only what lower-layer operations do.

Advantage:

1) Simplicity of construction and debugging.

Disadvantages:

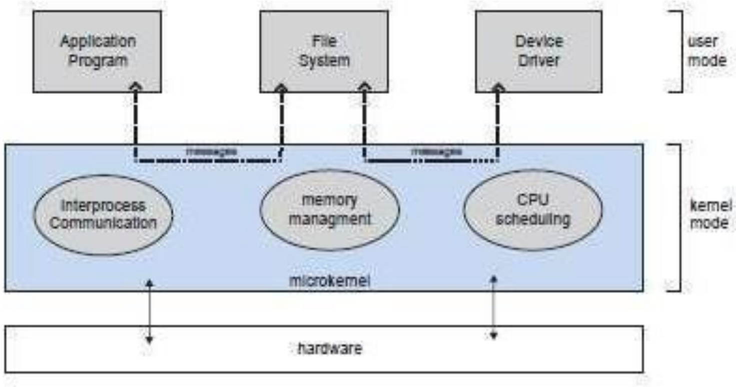
1) Less efficient than other types.
2) Appropriately defining the various layers.

MICRO-KERNEL APPROACH

Explain the advantages of Micro-kernel approach OS with neat diagram

Main function:

- ✓ To provide a communication facility between
  - client program and
  - various services running in user-space.
- ✓ Communication is provided by message passing
- ✓ All non-essential components are
  - removed from the kernel and
  - implemented as system- & user-programs.



Advantages:

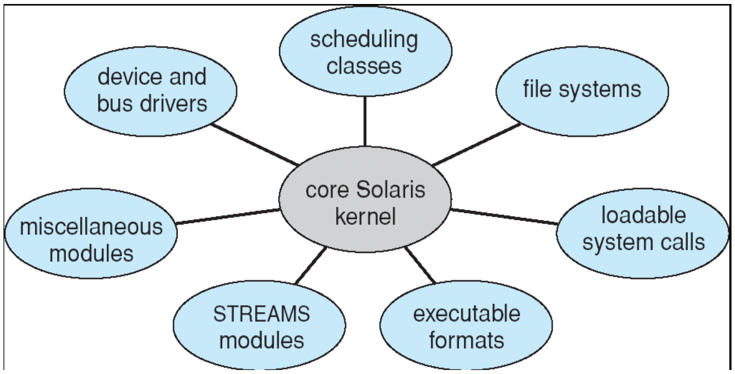
1) Ease of extending the OS. (New services are added to user space w/o modification of kernel).
2) Easier to port from one hardware design to another.
3) Provides more security & reliability.(If a service fails, rest of the OS remains untouched.).
4) Provides minimal process and memory management.

Disadvantage:

1. Performance decreases due to increased system function overhead
2. Performance overhead of user space to kernel space communication.

MODULAR-KERNEL APPROACH (MODULES)

Analyze the modular kernel approach with neat diagram.



Solaris loadable modules

The kernel has

→ set of core components and

→ dynamic links in additional services during boot time

Seven types of modules in the kernel

1. Scheduling classes
2. File systems
3. Loadable system calls
4. Executable formats
5. STREAMS modules
6. Miscellaneous
7. Device and bus drivers

The top layers include

→ application environments and

→ set of services providing a graphical interface to applications.

Kernel environment consists primarily of

→ Mach microkernel and

→ BSD kernel.

Mach provides

→ memory management;

→ support for RPCs & IPC and

→ thread scheduling.

BSD component provides

→ BSD command line interface

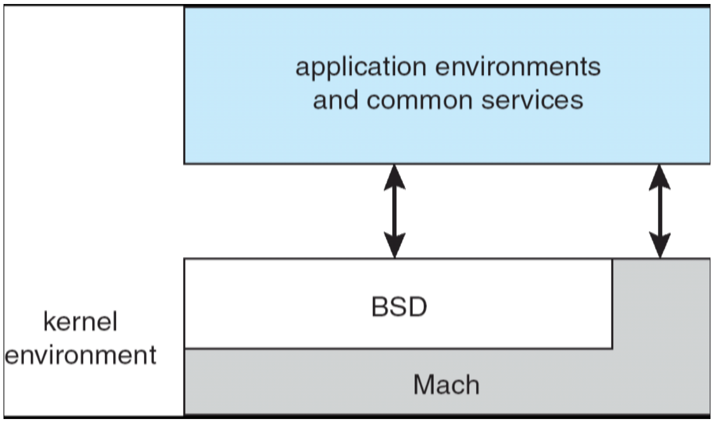
→ support for networking and file systems and

→ implementation of POSIX APIs

The kernel environment provides an I/O kit for development of

→ device drivers and

→ dynamic loadable modules (which Mac OS X refers to as kernel extensions).

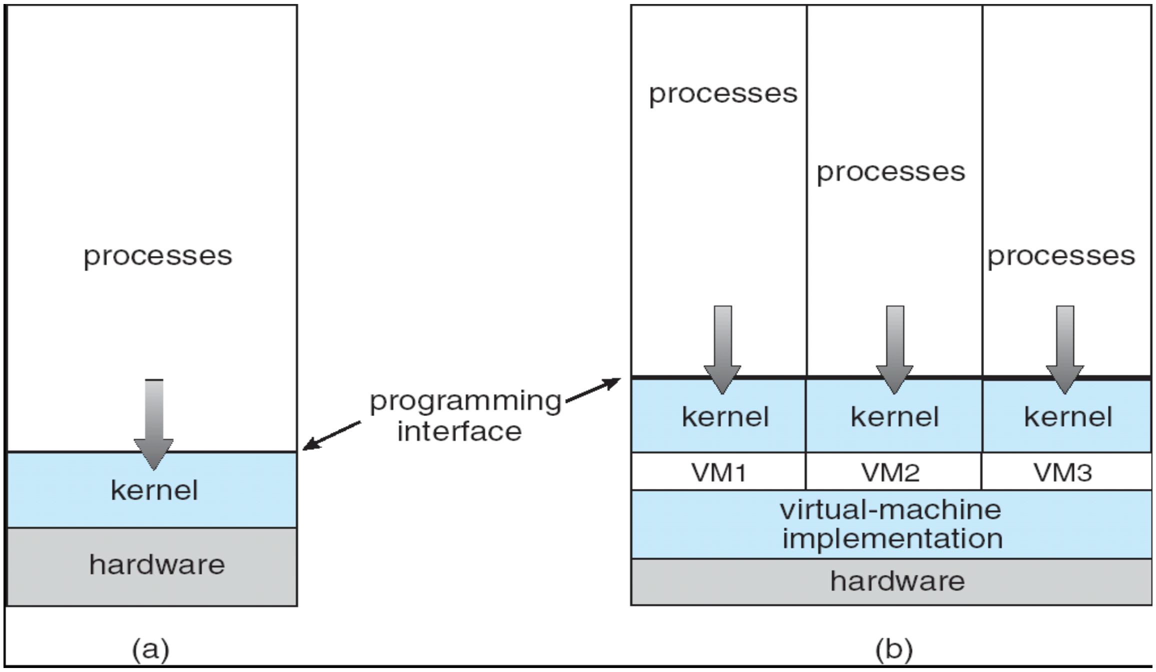


VIRTUAL MACHINES

********What are Virtual Machines? With neat diagram explain the concept of virtualization and its advantages.**

A **virtual machine** (VM) is a software program or operating system that not only exhibits the behavior of a, separate computer hardware, but is also capable of performing tasks such as running applications and programs like a separate computer.

- The fundamental idea behind the virtual machine is to abstract the hardware of a single computer (CPU, memory, disk drives, network interface cards) into several execution environments thereby creating an illusion that each separate execution environment is running its own private computer.
- The operating system creates the illusion of multiple processes, each executing on its own processor with its own (virtual) memory.
- Major difficulty is disk systems.



a) Non-virtual machine **b) Virtual machine**

The virtual-machine provides

- ✓ an interface that is identical to the underlying hardware.
- ✓ a (virtual) copy of the underlying computer to each process.

Problem:

- ✓ Virtual-machine software itself will need substantial disk space to provide virtual memory.

Solution: provide virtual disks that are identical in all respects except size.

Advantages:

1) Complete protection of the various system resources.
2) It is a perfect vehicle for OS‟s R&D.

Disadvantage:

1. Difficult to implement due to effort required to provide an exact duplicate to underlying machine.

IMPLEMENTATION OF VIRTUAL MACHINE

Explain how virtual machines are implemented.

- ✓ Though virtual machine concept is useful it is difficult to implement
- ✓ Much work is required to provide an exact duplicate of underlying machine
- ✓ The underlying machine has two modes: user mode and kernel mode
- ✓ Virtual machine software can run in kernel mode
- ✓ Virtual machine itself can run in user mode
- ✓ We have virtual user mode and virtual kernel mode with, transfer of control from virtual user mode to virtual kernel mode on virtual machine.
- ✓ The CPU is multi-programmed among many virtual machines.

BENEFIT OF CREATING VIRTUAL MACHINE

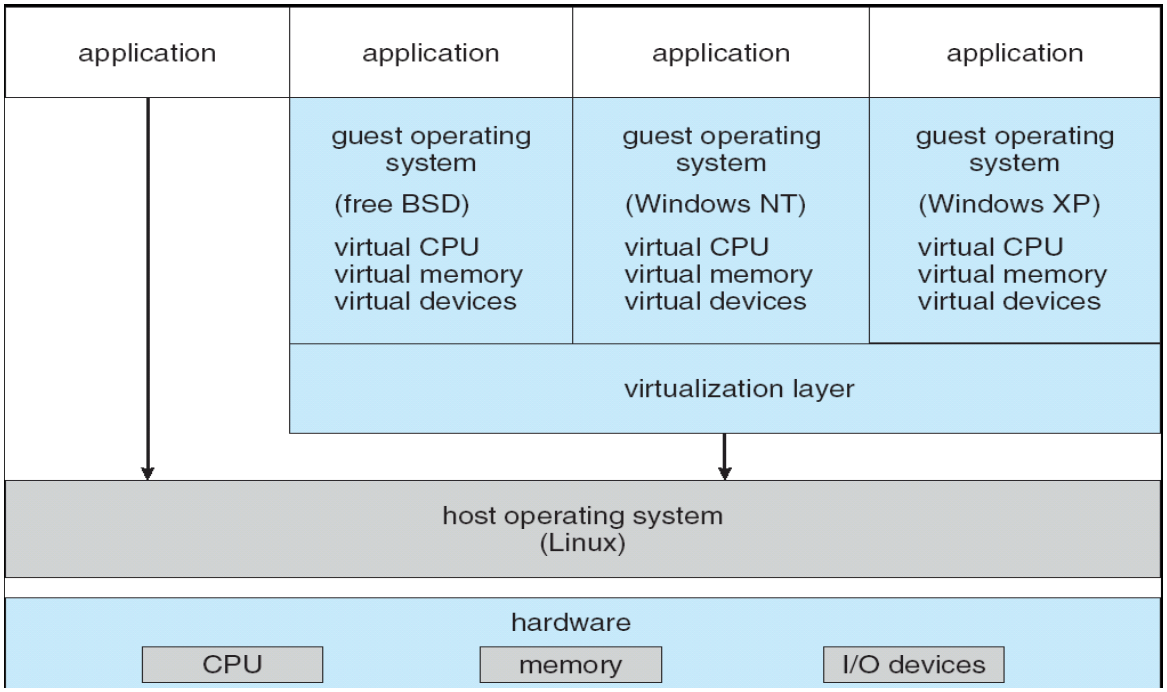
Explain the benefit of creating virtual machine

- ✓ All virtual machines share the same hardware yet run different execution environment
- ✓ Host system is protected from the virtual machine just as virtual machines are protected from each other. (virus)
- ✓ There is no direct sharing of resources
- ✓ However it is made possible to share the file-system volume and thus share files
- ✓ Network of virtual machines each of which can send information over the virtual communication network

- ✓ Virtual machine is a perfect vehicle of research and development
- ✓ The current operating system must be stopped and taken out of use while changes are made and tested. This period is called as system development time.
- ✓ Multiple operating systems can be running in developers workstation concurrently
- ✓ Data center use is system consolidation (run two or more separate system and run in virtual machine of one system)

VMWARE

*****Explain VM-WARE architecture of virtual machine with neat diagram



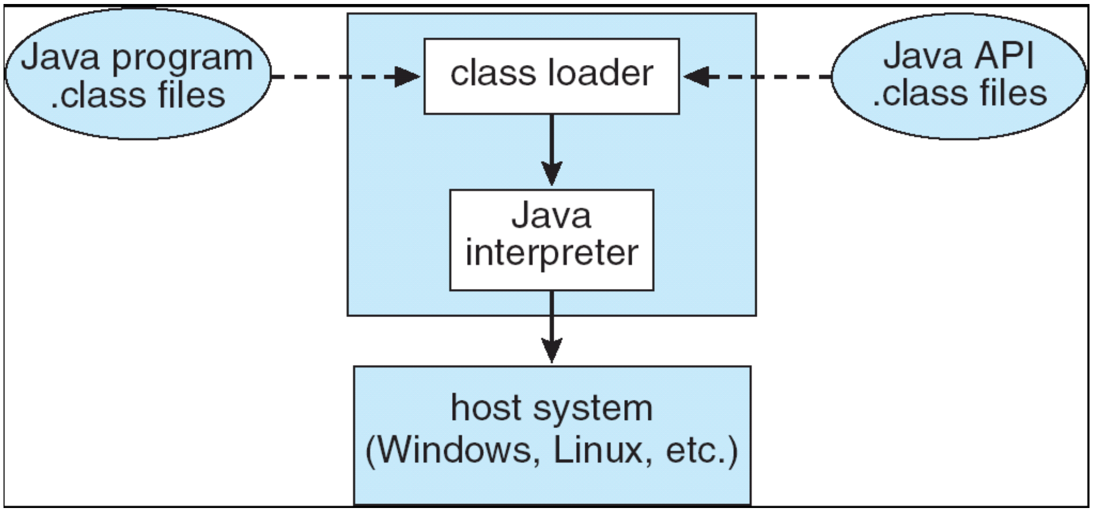
Key Idea: This virtualization tool run in user mode as an application on top of the operating system such as Windows or Linux and allows this host system to concurrently execute several different guest operating systems as independent virtual machines.

- VM-WARE abstracts intel 8086 and compatible hardware into isolated virtual machines
- Linux OS is running as the host OS: FreeBSD, Windows NT and Windows XP are running as guest OS
- The virtualization layer is the heart of VM-WARE, as it abstracts the physical hardware into isolated virtual machines running as guest operating systems.

- Each virtual machine has its own virtual CPU, memory, disk drives, network interface and so forth.
- An application could be tested concurrently in one physical computer with different OS using VMware.
- Programmer could test the application on a host operating system and on three guest operating systems with each system running as a separate virtual machine by using VMware

JAVA VIRTUAL MACHINE

Explain VM-WARE architecture of virtual machine with neat diagram



- ✓ Java provides the specification of java virtual machine
- ✓ Java objects are specified with class construct
- ✓ Java program consists of one or more classes
- ✓ For each java class the compiler produces a architectural neutral bytecode.
- ✓ JVM is a specification for an abstract computer
- ✓ Class loader and java interpreter
- ✓ Garbage collection
- ✓ JVM can be implemented on top of host OS, hardware
- ✓ Java interpreter interprets the bytecode operations one at a time
- ✓ A faster software technique is to use just-in-time(JIT) compiler