Study Material			
Department of AIML			
Course Code :	**BCS303**	**Semester / Year :**	**III / II**
Course Title :	**Operating Systems**	**Academic Year :**	**2024-25**

Important

Please refer to Text Book for detailed notes:

Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 8th edition, Wiley-India, 2015

Module 1

Unit 1

Introduction to operating systems, System structures: What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Special-purpose systems; Computing environments.

Introduction to Operating System

- What Operating Systems Do
- Computer-System Organization
- Computer-System Architecture
- Operating-System Structure
- Operating-System Operations
- Process Management
- Memory Management
- Storage Management
- Protection and Security
- Distributed Systems
- Special Purpose system
- Computing Environment

What is an Operating System

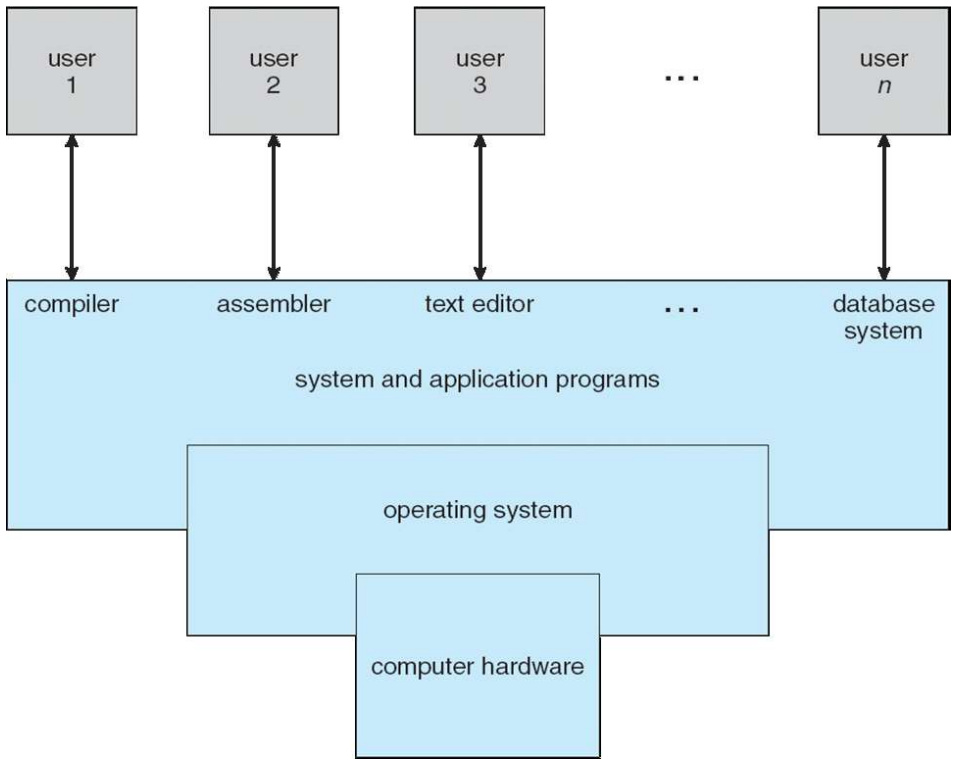
- A program that acts as an intermediary between a user of a computer and the computer hardware

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

Computer System Structure

- Computer system can be divided into four components:
  - Hardware – provides basic computing resources
    - CPU, memory, I/O devices
  - Operating system
    - Controls and coordinates use of hardware among various applications and users
  - Application programs – define the ways in which the system resources are used to solve the computing problems of the users
    - Word processors, compilers, web browsers, database systems, video games
  - Users
    - People, machines, other computers

Four Components of Computer System



What operating System Do?

- USER'S VIEW :
- Single User :
  - want convenience, **ease of use** and **good performance**
  - Don't care about **resource utilization**
  - **Optimized for Single User**
- Multi User :
  - Users Uses computer/terminal such as **mainframe** or **minicomputer** satisfying its request
  - Same computer/terminal accessed by other users through other terminals sharing resources and exchanging information.

What operating System Do?

- USER'S VIEW :
- Users of dedicate systems such as **workstations** have dedicated resources but frequently use shared resources from **servers**
- Handheld computers are resource poor, optimized for usability and battery life.poor performance due to power , speed and interface limitations
- Some computers have little or no user interface, such as embedded computers in devices and automobiles

What operating System Do?

- SYSTEM VIEW
  - OS is a **resource allocator**
    - Manages all resources
    - Decides between conflicting requests for efficient and fair resource use
  - OS is a **control program**
    - Controls execution of programs to prevent errors and improper use of the computer

Operating System Definition (Contd…)

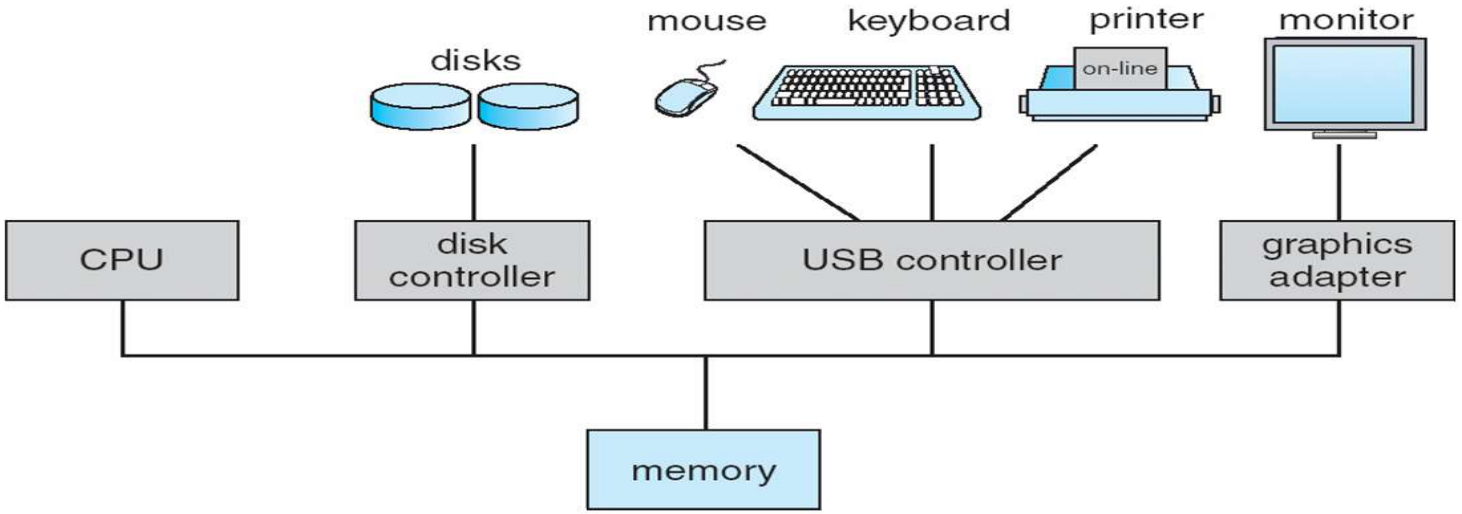
- No universally accepted definition
- "Everything a vendor ships when you order an operating system" is a good approximation
  - But varies wildly
- "The one program running at all times on the computer" is the **kernel**.
- Everything else is either
  - a system program , or
  - an application program.

Computer Startup

- **bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as **firmware**
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

Computer System Organization

- Computer-system operation
  - One or more CPUs, device controllers connect through common bus providing access to shared memory
  - Concurrent execution of CPUs and devices competing for memory cycles



Computer System Operations

- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller
- Device controller informs CPU that it has finished its operation by causing an interrupt

Common Functions of Interrupts

- OS starts executing first process **init** and wait for an event signaled by an interrupt from hardware /software.

- **Hardware** trigger an interrupt any time sending a signal to CPU through system bus

- **Software** trigger an interrupt executing special operation called **system call**

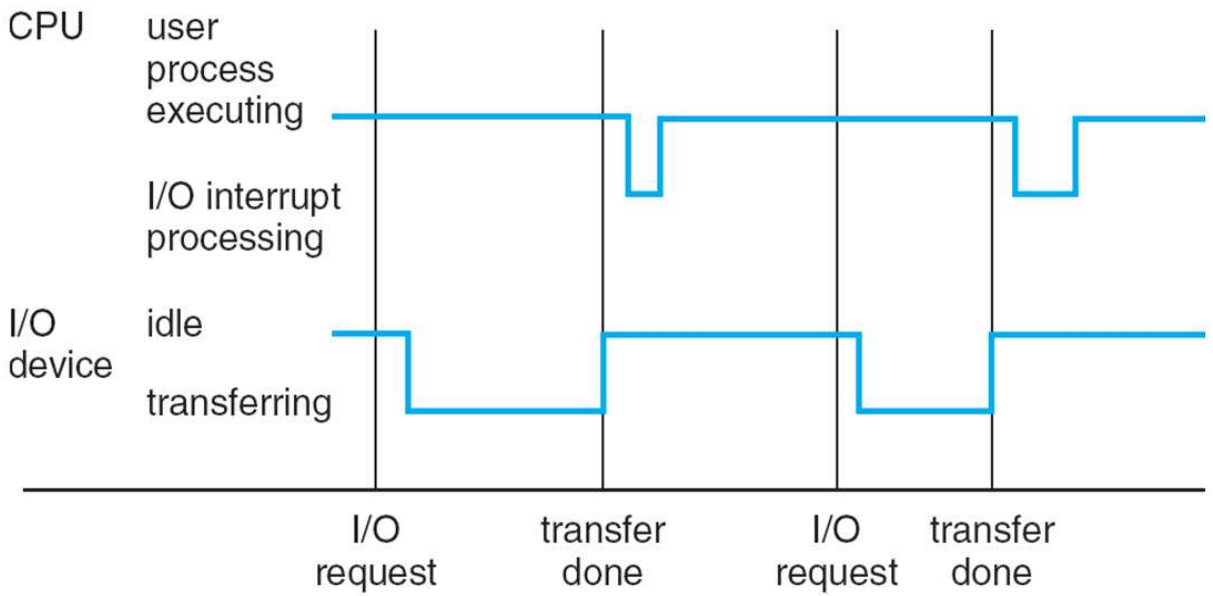
- A **trap** or **exception** is a software-generated interrupt caused either by an error or a user request

- An operating system is **interrupt driven**

Common Functions of Interrupts

- When CPU is interrupted ,it stop its operation & immediately transfer execution to fixed
- Fixed Location contain starting address where service routine for interrupt is located
- Interrupt service routine executes
- On completion CPU resumes interrupted computations

Interrupt Timeline



Common Functions of Interrupts

- Interrupts are important part of computer architecture .Each computer design has its own interrupt mechanism but several functions are common

- **Interrupt must transfer control to interrupt service routine**

- Straight forward method for handling this transfer would be to invoke a general routine to examine interrupt information i.e routine inturn it calls interrupt specific handler to handle interrupt quickly.

Common Functions of Interrupts

- Since predefined routines possible.

- **A table of pointer to interrupt routine can be used instead to provide necessary speed**

- The interrupt routine is called indirectly through the table with no intermediate routine needed.

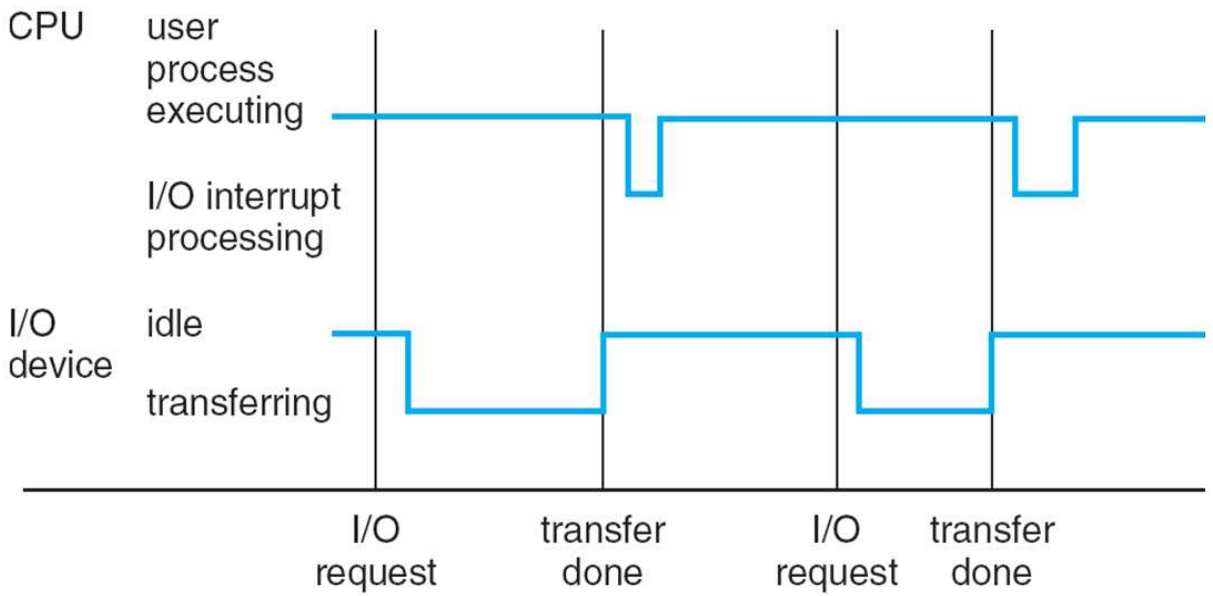
- Table of pointers stored in low memory .These location hold address of interrupt service routine for various devices called **Interrupt vector**

- Interrupt vector is indexed by unique device number given with interrupt request to provide address of interrupt service routine for interrupting device

Interrupt Handling

- The operating system preserves the state of the CPU by storing registers and the program counter
- Determines which type of interrupt has occurred:
  - **polling**
  - **vectored** interrupt system
- Separate segments of code determine what action should be taken for each type of interrupt

Interrupt Timeline



Storage Definition and Notations

The basic unit of computer storage is the **bit**.

A bit can contain one of two values, **0** and **1**.

All other storage in a computer is based on **collections of bits**.

Given enough bits, it is amazing how many things a computer can represent:
numbers, letters, images, movies, sounds, documents, and programs.

A **byte** is 8 bits, and on most computers it is the smallest convenient chunk of storage.

For example, most computers don't have an instruction to move a bit but do have one to move a byte.

Storage Definition and Notations

A less common term is **word**, which is a given computer architecture's native unit of data.

A word is made up of one or more bytes.

For example, a computer that has 64-bit registers and 64-bit memory addressing typically has 64-bit (8-byte) words.

A computer executes many operations in its native word size rather than a byte at a time.

Storage Definition and Notations

Computer storage, along with most computer throughput, is generally measured and manipulated in bytes and collections of bytes.

A **kilobyte**, or **KB**, is 1,024 bytes
a **megabyte**, or **MB**, is $1,024^2$ bytes
a **gigabyte**, or **GB**, is $1,024^3$ bytes
a **terabyte**, or **TB**, is $1,024^4$ bytes
a **petabyte**, or **PB**, is $1,024^5$ bytes

Computer manufacturers often round off these numbers and say that a megabyte is 1 million bytes and a gigabyte is 1 billion bytes. Networking measurements are an exception to this general rule; they are given in bits (because networks move data a bit at a time).

Storage Structure

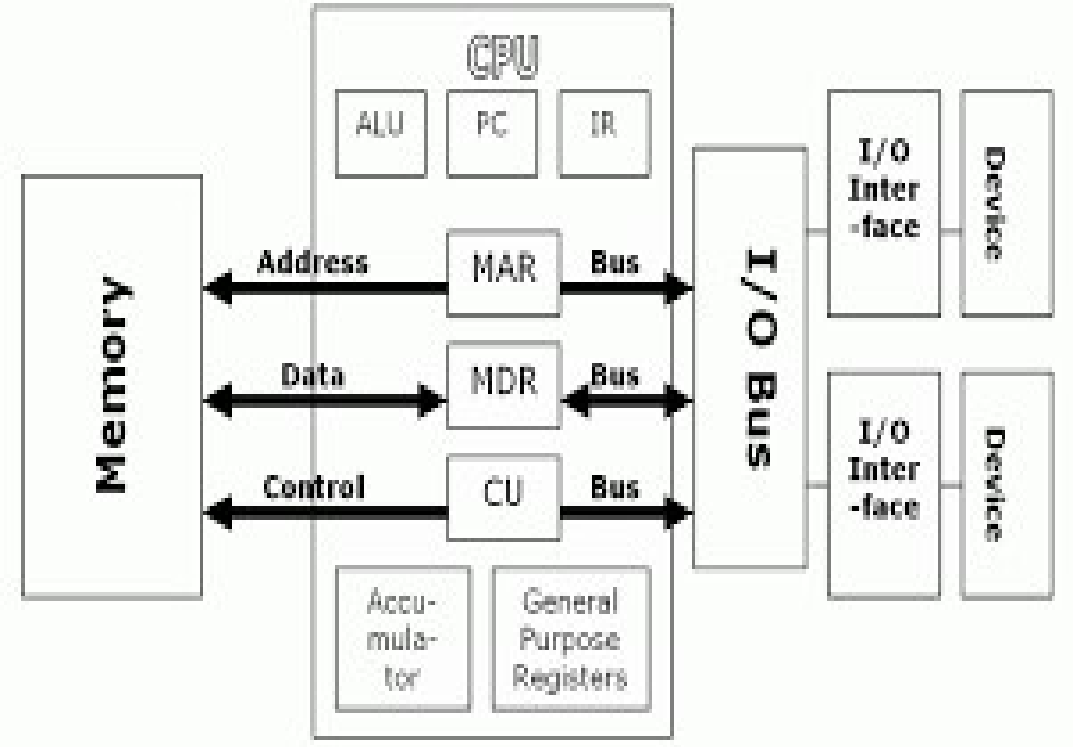
- Main memory – only large storage media that the CPU can access directly
  - **Random access (RAM)-**General purpose computers run their programs from rewritable memory
  - Typically **volatile**,
  - Implemented in a semiconductor technology called **Dynamic Random Access memory**

- Read only Memory (ROM)
  - **Not Rewritable memory**
  - Only **Static Programs** are stored
  - Typically **Non-volatile**

Storage Structure

- EEPROM– Electrically Erasable Programmable Read only Memory
  - **Non-Volatile** ROM enables individual bytes of data to be erased and reprogrammed
  - Contains **Static Programs**
  - **Smartphones** have EEPROM to **store** their **factory –installed programs**
- Memory – Array of words which has its own address
- Load and Store instruction is used to specify memory addresses
- Load – moves word from main memory- internal registers within CPU
- Store – moves word from internal registers- main memory
- CPU automatically loads instruction from main memory for execution

Storage Structure



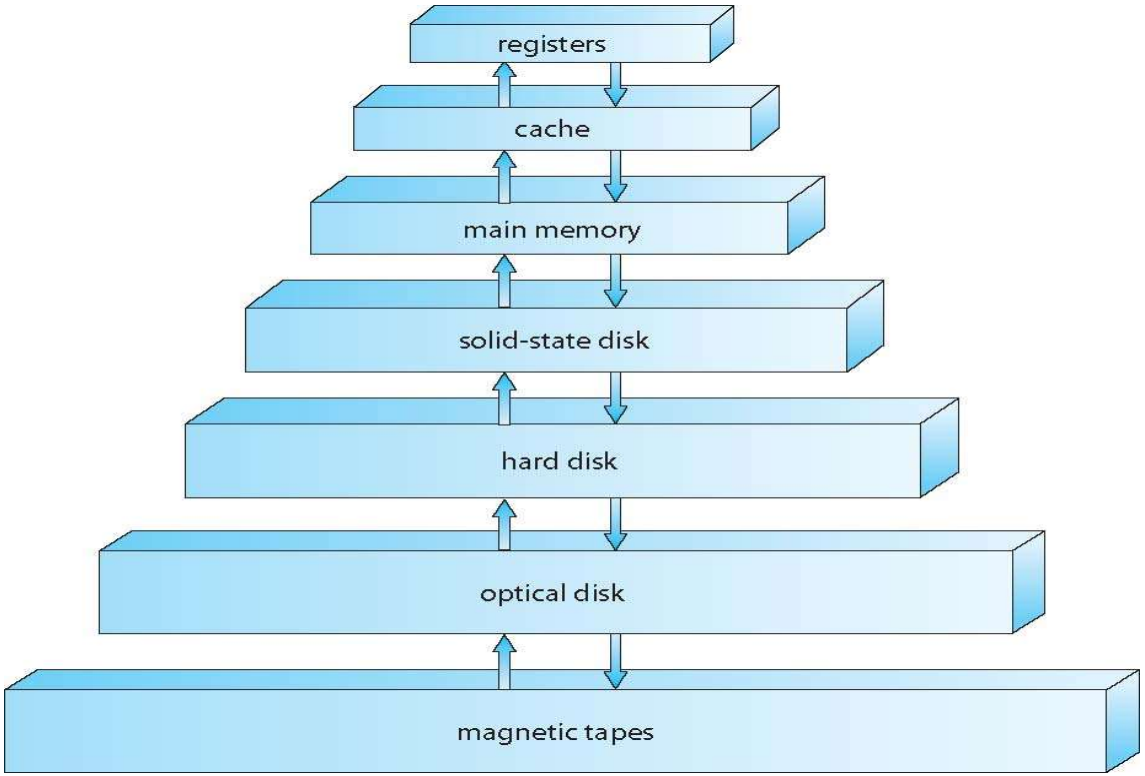
Storage Structure

- Main memory –
  - Typically **volatile**,it loses its content when power is turned off
  - **Main Memory** is smaller to store all needed programs data permanently
- Secondary storage – extension of main memory that provides large **nonvolatile** storage capacity
- Hold Large quantities of data
- Magnetic Disk :
    - Provide storage for both programs and data
    - Most programs (system & application) are stored in disk untill they are loaded in memory
    - Used both as programs and desination of processing

Storage Structure

- Hard 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
- **Solid-state disks** – faster than hard disks, nonvolatile
  - Various technologies
  - Becoming more popular

Storage Device - Hierarchy



Storage Hierarchy

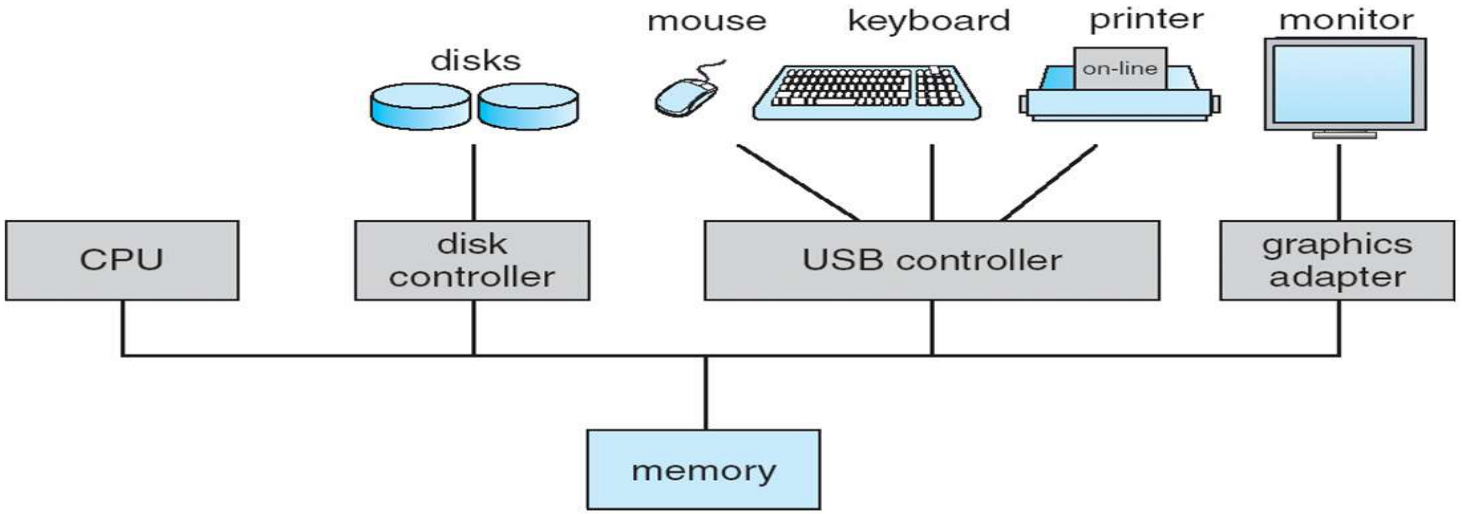
- Storage systems organized in hierarchy
  - Speed
  - Cost
  - Volatility
- **Caching** – copying information into faster storage system; main memory can be viewed as a cache for secondary storage
- **Device Driver** for each device controller to manage I/O
  - Provides uniform interface between controller and kernel

Caching

- Important principle, performed at many levels in a computer (in hardware, operating system, software)
- Information in use copied from slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there
  - If it is, information used directly from the cache (fast)
  - If not, data copied to cache and used there
- Cache smaller than storage being cached
  - Cache management important design problem
  - Cache size and replacement policy

I/O Structure

- General Purpose Computer System consist of one or more CPUs and multiple device controllers connect through common bus
- Each device controller is incharge of specific type of device



I/O Structure

- Device Driver : A device driver is a type of software programming that assists in connecting with various types of operating systems.

- Device Controller : A device controller is a type of hardware programming that acts as a bridge between OS in a computer system

I/O Structure

- Depend on controller ,more than one device can be attached (seven or more devices can be attached to small computer system interface(SCSI) controller
- A device controller maintain local buffer storage and special purpose registers
- The device controller is responsible for moving data between peripheral devices that it controls and its local buffer storage
- Operating system have device driver for each device controller
- Device driver understands device controller and it provide interface to the device and OS

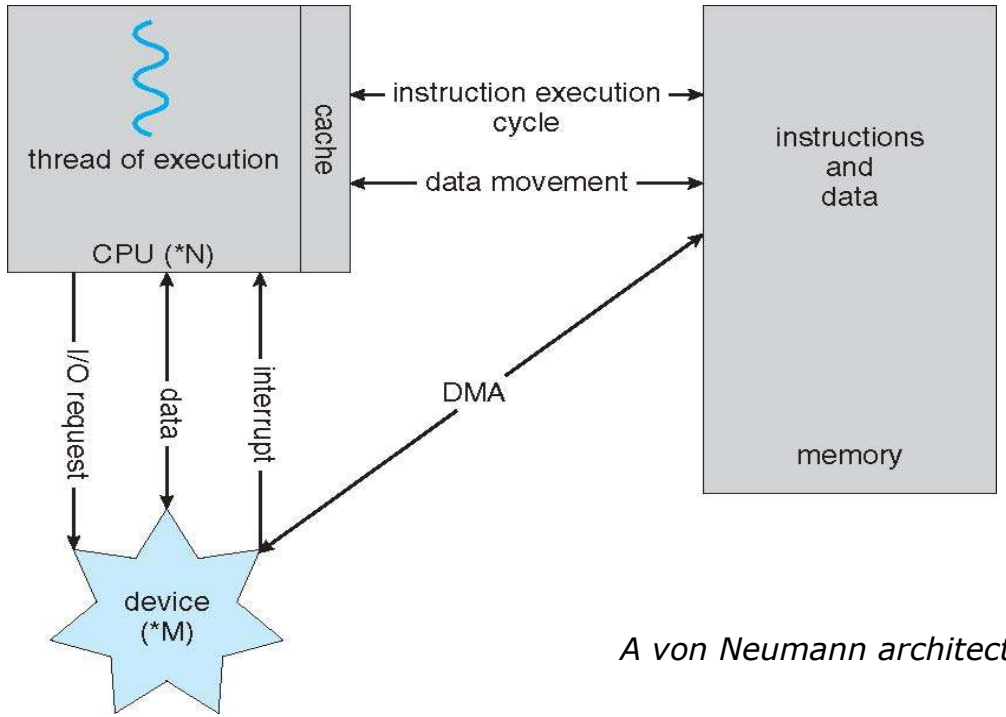
I/O Structure

- For I/O operations ,device driver loads the appropriate registers within device controller
- A device controller examines content of registers to determine which actions to take (Read a character from keyboard)
- Controller transfer data from device to its local buffer
- Once transfer of data complete ,device controller inform device driver via interrupt about completion of operation
- Device driver then return control to OS
- Interrupt driven I/O is suitable for moving small amount of data but produces high overhead for bulk data movement.

Direct Memory Access Structure

- After setting up buffers, pointers , and counters for I/O device, the device controller transfers entire block of data directly to and from its own buffer storage to without CPU intervention
- Only one interrupt is generated per block, rather than the one interrupt per byte

How a Modern Computer Works



A von Neumann architecture

1. High End systems rather than using bus architecture , switch is used
2. Multiple components communicate concurrently rather than completing for cycles on shared bus

Computer –System Architecture

- **Single processor Syatems** If systems contain only one single general-purpose processor (CPU).
  - Ex PDA ,Mainframe

- **CPU** capable of executing GPIS (General Purpose Instruction Set) and Instruction from user processes
  - on **mainframe ,**They come in the form of I/O processors that move data rapidly among components of systems.

- **All systems** have other special –purpose processor
  - Device specific processor **,ex** Disk , keyboard ,Graphics controller
  - Run limited instruction set

Computer –System Architecture

- Do not run User processes
- Managed by OS (OS send information about their next task and monitor their status) Ex: PC's contain microprocessor in the keyboard to convert keystroke into codes to be sent to CPU)
- Use of special purpose register is common and doesnot turn single processor system into multiprocessor

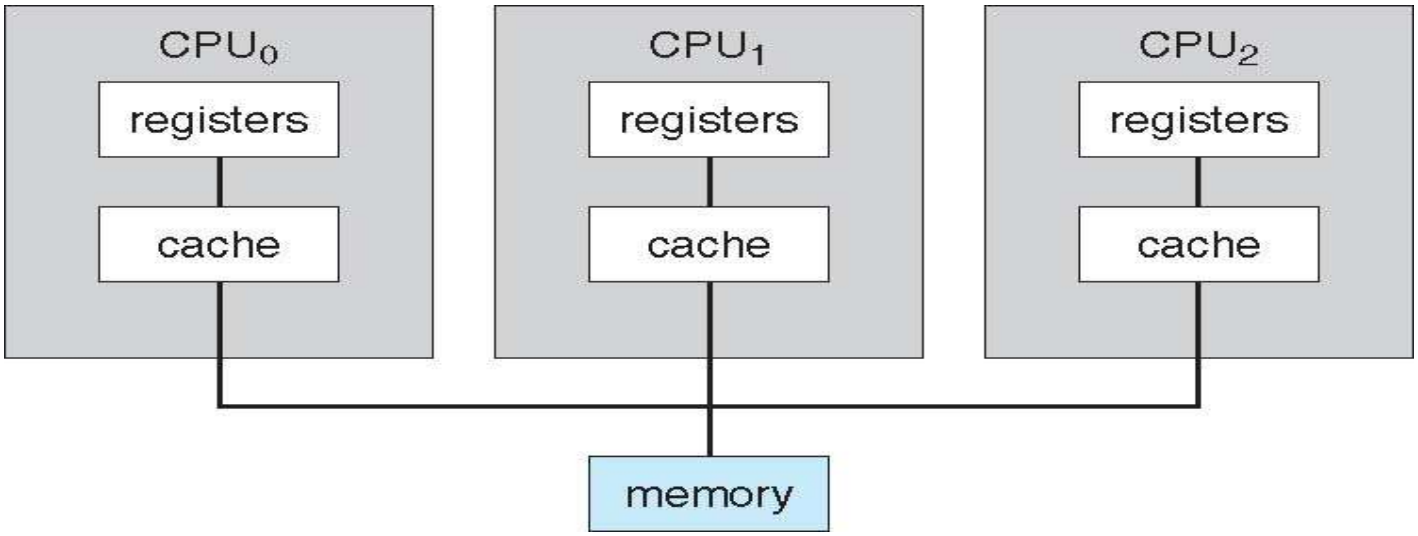
Computer –System Architecture

- **Multiprocessors** systems growing in use and importance
  - Contain two or more processors in close communication ,sharing computer bus ,clock ,memory and peripheral devices
  - Also known as **parallel systems**, **tightly-coupled systems**
  - Advantages include:
    1. **Increased throughput**
    2. **Economy of scale**
    3. **Increased reliability** – graceful degradation or fault tolerance

Computer –System Architecture

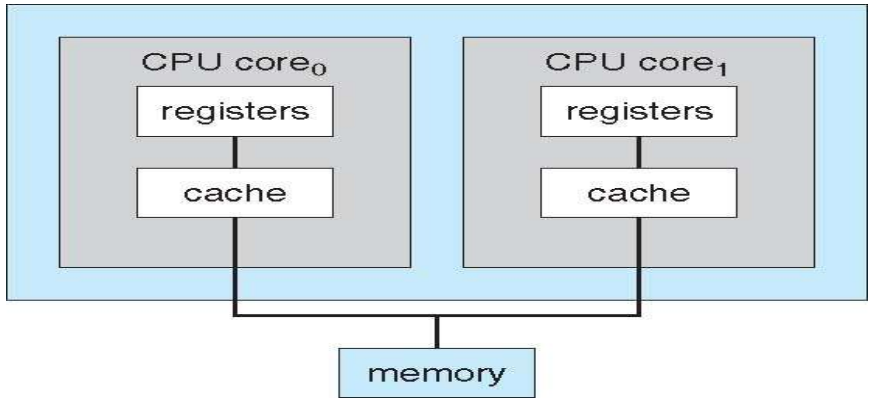
- Two types:
  1. **Asymmetric Multiprocessing** – each processor is assigned a specific task.
     - Master –slave relationship
     - Master controls the system
     - Other processor look to master for instruction or predefined task
  2. **Symmetric Multiprocessing** – each processor performs all tasks within OS
     - Peers processors
     - No master slave relationship between processors
     - Each processor has its own set of registers, local cache and shared physical memory

Symmetric Multiprocessing Architecture



A Dual – Core Design

- Core : A small Processor built into big CPU or CPU Socket
- Multiple computing **cores** on a single chip – Multiprocessor chip
  - Efficient than multiple chips with single core , on chip communication is faster than between chip communication
  - Uses less power than multiple single core chips
  - Well suited for Database and Web Server



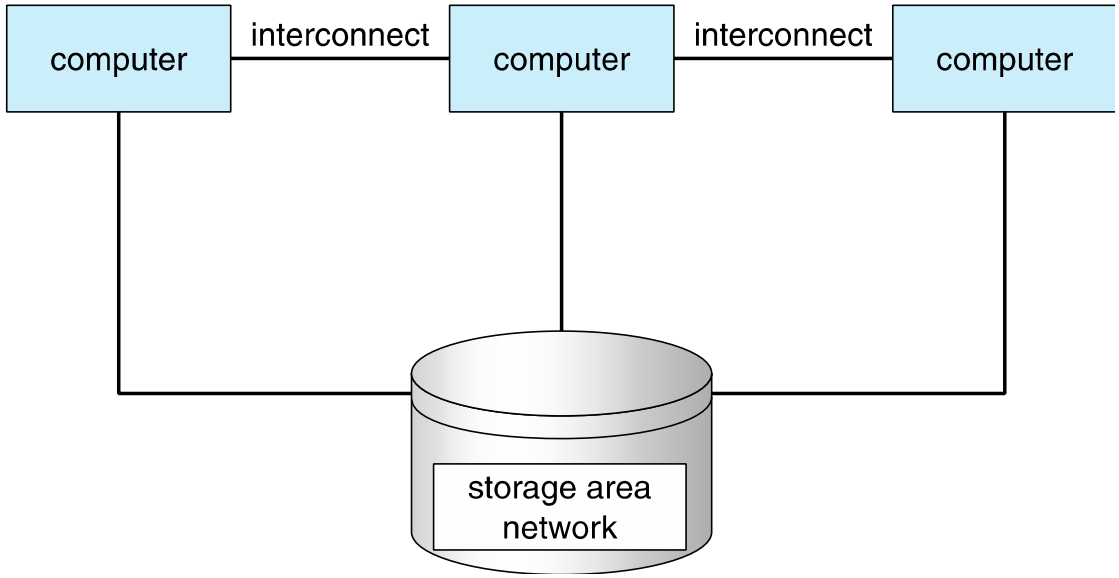
A Dual – Core Design

- **Blade Servers** are recent developments in which multiple processor board,IO boards, networking boards placed in same chasis
- Each Blade processor boards boots independently and run its OS.

Clustered Systems

- Like multiprocessor systems, but multiple systems working together
  - Usually sharing storage via a **local-area network (LAN)**
  - Provides a **high-availability** service which survives failures
    - **Asymmetric clustering** has one machine in hot-standby mode
    - Hot-Stand by mode does nothing but monitor active server.
    - **Symmetric clustering** has multiple nodes running applications, monitoring each other
  - Some clusters are for **high-performance computing (HPC)**
    - Applications must be written to use **parallelization**

Clustered Systems



Operating System Structure

- **Multiprogramming** (**Batch system**) needed for efficiency

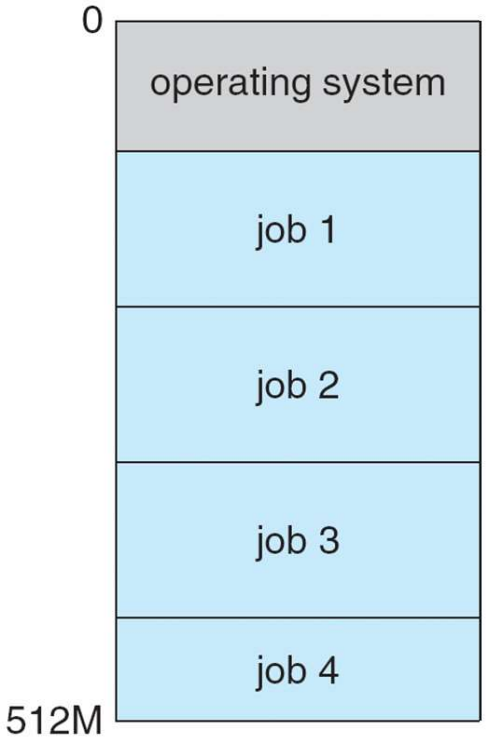
- Single user cannot keep CPU and I/O devices busy at all times
- Multiprogramming organizes jobs (code and data) so CPU always has one to execute
- A subset of total jobs in system is kept in memory
- One job selected and run via **job scheduling**
- When it has to wait (for I/O for example), OS switches to another job

Operating System Structure

- **Timesharing** (**multitasking**) is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing

- **Response time** should be < 1 second
- Each user has at least one program executing in memory → **process**
- If several jobs ready to run at the same time → **CPU scheduling**
- If processes don't fit in memory, **swapping** moves them in and out to run
- **Virtual memory** allows execution of processes not completely in memory

Memory Layout for Multiprogrammed System



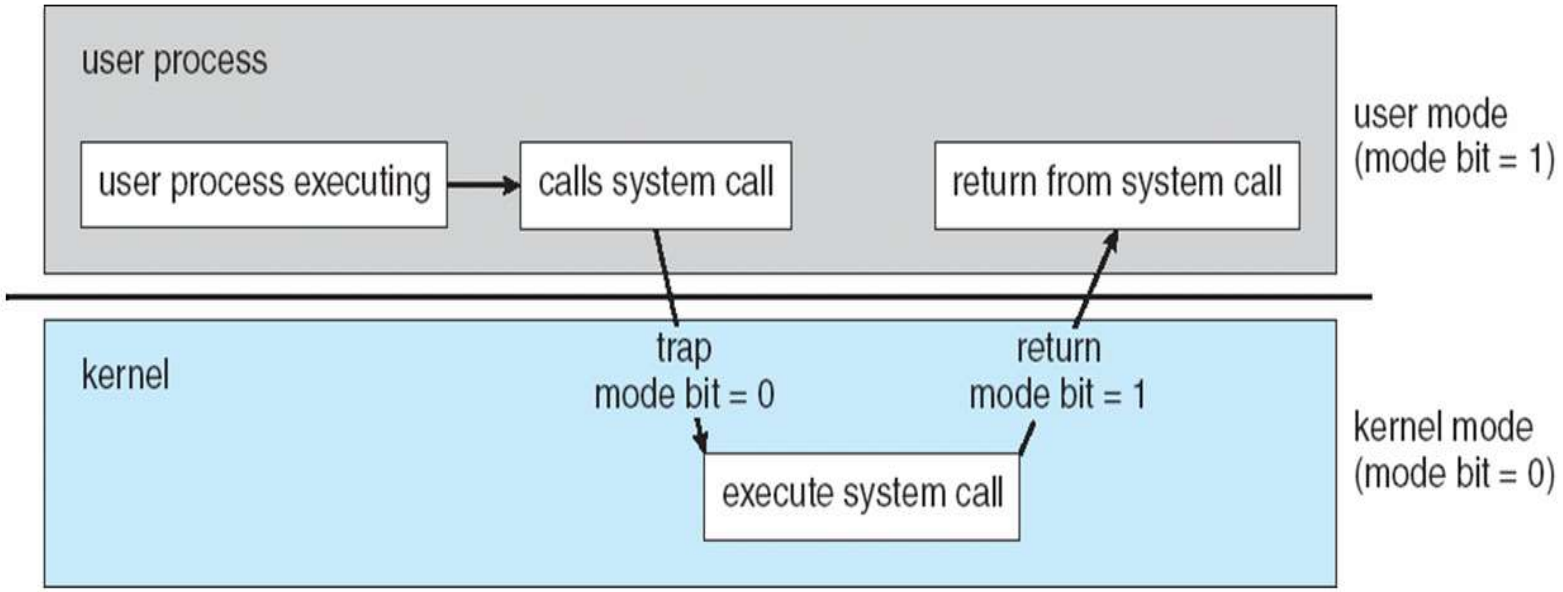
Operating System Operations

- **Interrupt driven** (hardware and software)
  - Hardware interrupt by one of the devices
  - Software interrupt (**exception** or **trap):**
    - Software error (e.g., division by zero)
    - Request for operating system service
    - Other process problems include infinite loop, processes modifying each other or the operating system

Operating System Operations(Contd…)

- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode**
  - **Mode bit** provided by hardware
    - Provides ability to distinguish when system is running user code or kernel code
    - Some instructions designated as **privileged**, only executable in kernel mode
    - System call changes mode to kernel, return from call resets it to user
- Increasingly CPUs support multi-mode operations
  - i.e. **virtual machine manager** (**VMM**) mode for guest **VMs**

Transition from User to Kernel Mode



Timer

- Timer to prevent infinite loop / call system services/fails to return to OS
  - Timer is set to interrupt the computer after some time period.It can be fixed time period (1/60 seconds)or variable(1 millisecond to 1 second)
  - Variable timer is implemented by a fixed rate clock and a counter)Keep a counter that is decremented by the physical clock.
  - Operating system set the counter and every time the clock ticks counter is decremented
  - When counter zero generate an interrupt

Timer

- Before returning control to user , operating system ensures that timer is set interrupt
- If timer interrupts ,control transfers automatically to the operating system.

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
  - CPU, memory, I/O, files
  - Initialization data
- Process termination requires reclaim of any reusable resources

Process Management

- 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
  - Concurrency by multiplexing the CPUs among the processes / threads

Process Management Activities

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

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

Memory Management

- To execute a program all (or part) of the instructions must be in memory
- All (or part) of the data that is needed by the program must be in memory.
- Memory management determines what is in memory and when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed

Storage Management

- OS provides uniform, logical view of information storage
  - The OS Abstracts from the physical properties of its storage devices to define a logical storage unit - **file**
  - The OS maps files onto physical media and access these files Via the Storage device (i.e., disk drive, tape drive)
    - Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)

Storage Management

File-System management

- A File is a collection of related information defined by its creator
- File represent programs (source and object forms) and data
- Data files may be numeric ,alphabetic ,alphanumeric or binary
- The operating system implements abstract concept of a file by managing mass storage media such as tapes and disks and devices that control them

Storage Management

- Files usually organized into directories
- Access control on most systems to determine who can access what
- OS activities include
  - Creating and deleting files
  - Creating and deleting directories to organize files
  - Supporting Primitives for manipulating files and directories
  - Mapping files onto secondary storage
  - Backing up files 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
- Proper management is of central importance
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- OS activities
  - Free-space management
  - Storage allocation
  - Disk scheduling

Mass Storage Management

- Some storage need not be fast
  - Tertiary storage includes optical storage, magnetic tape
  - Still must be managed – by OS or applications
  - Varies between WORM (write-once, read-many-times) and RW (read-write)

Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

Movement between levels of storage hierarchy can be explicit or implicit

Protection and Security

- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
- **Security** – defense of the system against internal and external attacks
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service
- Systems generally first distinguish among users, to determine who can do what
  - User identities (**user IDs**, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (**group ID**) allows set of users to be defined and controls managed, then also associated with each process, file
  - **Privilege escalation** allows user to change to effective ID with more rights

Distributed System

- Distributed computing
  - Collection of separate, possibly heterogeneous, systems networked together to provide users access to various resources that the system maintains
  - Access to a shared resources increases ***computation speed*** ,***functionality*** ,***data availability*** and ***reliability***.
    - **Network** is a communications path, **TCP/IP** most common
      - ***Local Area Network*** (**LAN**)
      - ***Wide Area Network*** (**WAN**)
      - ***Metropolitan Area Network*** (**MAN**)
      - ***Personal Area Network*** (**PAN**)
  - **Network Operating System** provides features between systems across network
    - Communication scheme allows systems to exchange messages
    - Illusion of a single system

Special Purpose System

Real Time Embedded System

- Real-time embedded systems most prevalent form of computers found everywhere from car engines to manufacturing
- Consist of very specific tasks robots to DVD's and microwave ovens
- These systems run on their primitives and so operating system provide limited features.
- These system have little User interface preffering to spend their time ,managing and monitoring hardware devices ,such as automobile engines and robotic arm

Special Purpose System

Real Time Embedded System

- Real-time embedded systems vary considerably
- Some are general purpose computers running standard OS such as UNIX with special purpose applications to implement functionality
- Some hardware devices with special purpose embedded OS providing desired functionality
- Real-time OS has well-defined fixed time constraints
  - Processing ***must*** be done within constraint
  - Correct operation only if constraints met

Special Purpose System

Multimedia System

- **Multimedia** describes a wide range of applications
- It includes audio files such as MP3,DVD movies, video conferencing and short video clips of movie preview or news stories downloaded over the internet.
- Multimedia applications include live webcasts (broadcasting over www) of speeches or supporting events and even live webcam
- Multimedia applications need not be either audio or video ,rather combination of both (movie consist of separate audio and video tracks)
- Multimedia applications be delivered only to desktop personal computer Increasingly ,they are being directed toward smaller devices

Special Purpose System

Handheld Systems:

- **Handheld systems** include personal digital assistants (PDA's) such as Palm and Pocket-PC's and cellular phones –uses special purpose embedding operating systems
- Developers of handheld systems and application face many challenges ,due to the limited size of such devices
  - PDA is about 5 inches in height and 3 inches in width and its weight less than one-half pound
  - Handheld devices have small amounts of memory ,slow processors and small display screens

Special Purpose System

Handheld Systems:

- **Limitations of Handheld systems**

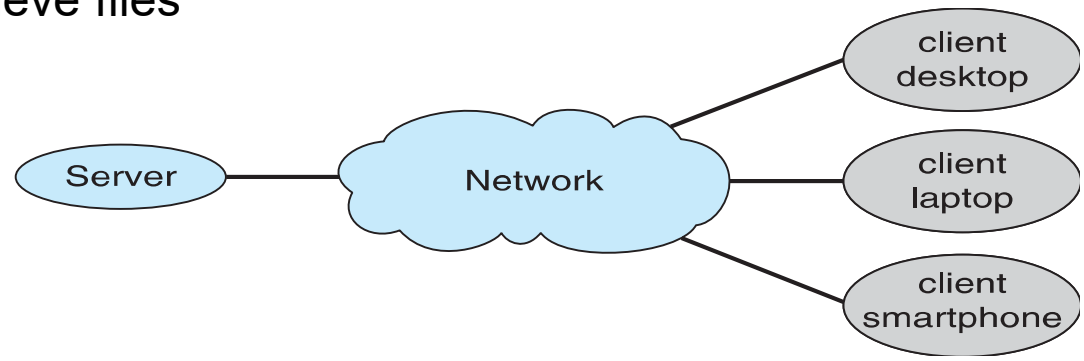
  - The amount of physical memory depends on the device ,typically between 1MB to 1GB
  - Speed of processor – runs at a fraction of the speed of processor .Faster processor requires a larger battery
  - Lack of physical space limits input methods to small keyboards, handwriting recognition, or small screen-based keyboards.

Computing Environments - Traditional

- Stand-alone general purpose machines
- But blurred as most systems interconnect with others (i.e., the Internet)
- **Portals** provide web access to internal systems
- **Network computers** (**thin clients**) are like Web terminals
- Mobile computers interconnect via **wireless networks**
- Networking becoming ubiquitous – even home systems use **firewalls** to protect home computers from Internet attacks

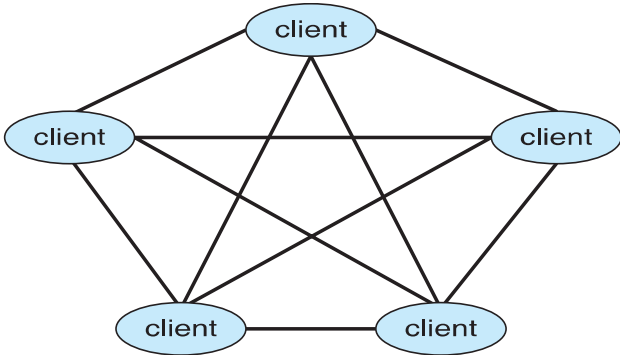
Computing Environments – Client-Server

- Client-Server Computing
  - Dumb terminals supplanted by smart PCs
  - Many systems now **servers**, responding to requests generated by **clients**
    - **Compute-server system** provides an interface to client to request services (i.e., database)
    - **File-server system** provides interface for clients to store and retrieve files



Computing Environments - Peer-to-Peer

- Another model of distributed system
- P2P does not distinguish clients and servers
  - Instead all nodes are considered peers
  - May each act as client, server or both
  - Node must join P2P network
    - Registers its service with central lookup service on network, or
    - Broadcast request for service and respond to requests for service via ***discovery protocol***
  - Examples include Napster and Gnutella, **Voice over IP** (**VoIP**) such as Skype



Computing Environments – Web-Based Computing

- Web has become ubiquitous , leading to more access by a wider variety of devices
- PC's most prevalent access devices with workstation ,handheld PDA's and even cell phones providing access.
- Web computing has increases the emphasis on networking ,Devices are networked through wired or wireless technology
- Implementation of web-based computing given rise to new categories of devices such as load balancers which distribute network connection among pool of servers