

SSCJUNCTION.COM

SSCJUNCTION.COM

IBPS IT OFFICER EXAM CAPSULE

Overview on SQL Queries

- **DML - Data Manipulation Language:**

Command	Description
SELECT	Retrieves certain records from one or more tables
INSERT	Creates a record
UPDATE	Modifies records
DELETE	Deletes records

- **DCL - Data Control Language:**

Command	Description
GRANT	Gives a privilege to user
REVOKE	Takes back privileges granted from user

- **DDL - Data Definition Language:**

Command	Description
CREATE	Creates a new table, a view of a table, or other object in database
ALTER	Modifies an existing database object, such as a table.
DROP	Deletes an entire table, a view of a table or other object in the database.

- Dr. Edgar F. "Ted" Codd of IBM is known as the father of relational databases. He described a relational model for databases.
- SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in relational database.
- **Primary and Unique Key difference:**
>>Both primary key and unique enforce uniqueness of the column on which they are defined. But by default primary key creates a clustered index on the

column, where are unique creates a non-clustered index by default. Another major difference is that, primary key doesn't allow NULLs, but unique key allows one NULL only.

- Difference between **GROUP BY** and **HAVING** Clause
>>Specifies a search condition for a group or an aggregate. HAVING can be used only with the SELECT statement. HAVING is typically used in a GROUP BY clause. When GROUP BY is not used, HAVING behaves like a WHERE clause. Having Clause is basically used only with the GROUP BY function in a query. WHERE Clause is applied to each row before they are part of the GROUP BY function in a query.
- Difference between a **local** and a **global** variable:
>>A local temporary table exists only for the duration of a connection or, if defined inside a compound statement, for the duration of the compound statement.
A global temporary table remains in the database permanently, but the rows exist only within a given connection. When connections are closed, the data in the global temporary table disappears.
- Primary keys are the unique identifiers for each row. They must contain unique values and cannot be null. Due to their importance in relational databases, Primary keys are the most fundamental of all keys and constraints. A table can have only one Primary key.
- Foreign keys are both a method of ensuring data integrity and a manifestation of the relationship between tables.
- SQL Profiler is a graphical tool that allows system administrators to monitor events in an instance of Microsoft SQL Server. You can capture and save data about each event to a file or SQL Server table to analyze later.
- SQL Server agent plays an important role in the day-to-day tasks of a database administrator (DBA).
- Log shipping is the process of automating the backup of database and transaction log files on a production SQL server, and then restoring them onto a standby server.

- Difference between a **Local and a Global temporary table**:

>>A local **temporary** table exists only for the duration of a connection or, if defined inside a compound statement, for the duration of the compound statement.

>>A global temporary table remains in the database **permanently**, but the rows exist only within a given connection. When connection is closed, the data in the global temporary table disappears.

- STUFF function is used to overwrite existing characters.

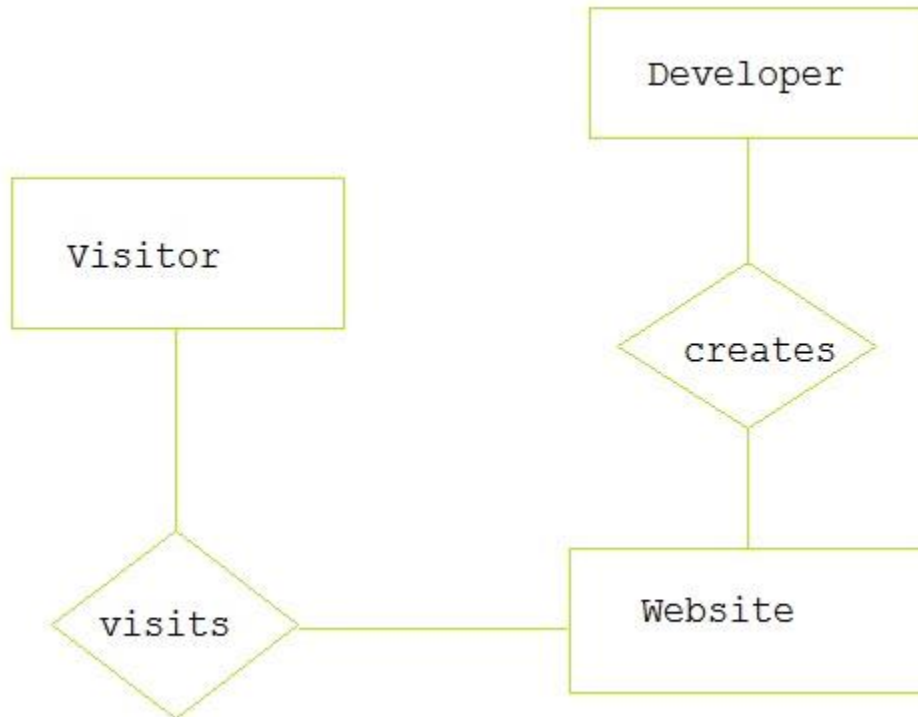
NORMALISATION

- **Normalization** is a technique of organizing data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form by removing duplicated data from the relation tables.
- Normalization is used for mainly two purpose:
 - Eliminating redundant (useless) data.
 - Ensuring data dependencies make sense i.e. data is logically stored.
- **Problem Without Normalization** :
 - Without Normalization, it becomes difficult to handle and update the database, without facing data loss. Insertion, Updation and Deletion Anomalies are very frequent if Database is not normalized.
- **Updation Anomaly**: To update address which occurs twice or more than twice in a table, we will have to update Address column in all the rows, else data will become inconsistent.
- **Insertion Anomaly**: Suppose for a new admission, we have a Student id (S_id), name and address of a student but if student has not opted for any subjects yet then we have to insert **NULL** there, leading to Insertion Anomaly.
- **Deletion Anomaly** has only one subject and temporarily he drops it, when we delete that row, entire student record will be deleted along with it.





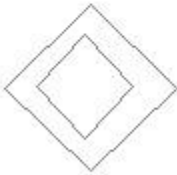
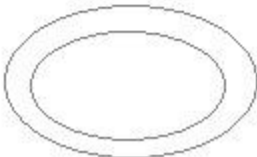
- Normalization rule are divided into following normal form.
 - 1.First Normal Form
 - 2.Second Normal Form
 - 3.Third Normal Form
 - 4.BCNF
- **First Normal Form (1NF)** :As per First Normal Form, no two Rows of data must contain repeating group of information i.e. each set of column must have a unique value, such that multiple columns cannot be used to fetch the same row. Each table should be organized into rows, and each row should have a primary key that distinguishes it as unique. The Primary key is usually a single column, but sometimes more than one column can be combined to create a single primary key.
- **Second Normal Form (2NF)**: As per the Second Normal Form there must not be any partial dependency of any column on primary key. It means that for a table that has concatenated primary key, each column in the table that is not part of the primary key must depend upon the entire concatenated key for its existence. If any column depends only on one part of the concatenated key, then the table fails Second normal form.
- **Third Normal Form (3NF)** :Third Normal form applies that every non-prime attribute of table must be dependent on primary key. The *transitive functional dependency* should be removed from the table. The table must be in Second Normal form
- **Boyce and Codd Normal Form (BCNF)**: Boyce and Codd Normal Form is a higher version of the Third Normal form. This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF.

E-R Diagrams

- ER-Diagram is a visual representation of data that describes how data is related to each other.



Symbols and Notations:

	represents
	Entity
	relationship
	attribute
	weak entity
	weak entity relationship
	Multivalued attribute

- Components of E-R Diagram

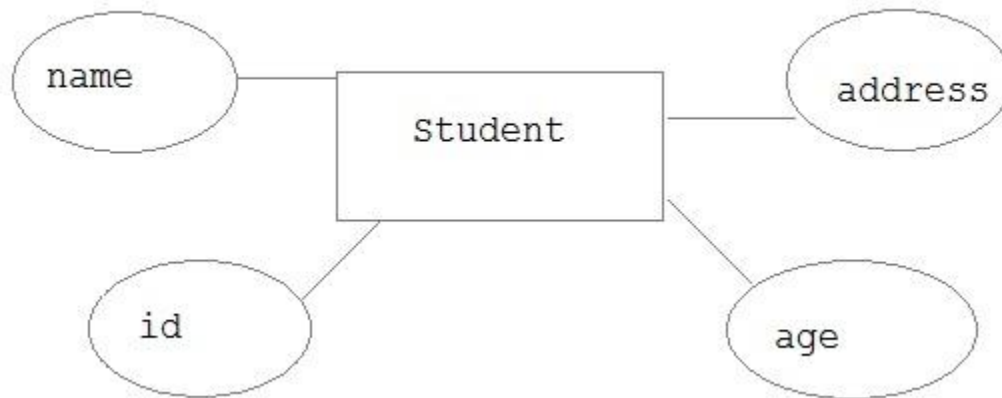
- The E-R diagram has three main components.
- Entity: An Entity can be any object, place, person or class. In E-R Diagram, an entity is represented using rectangles. Consider an example of an Organization. Employee, Manager, Department, Product and many more can be taken as entities from an Organization.



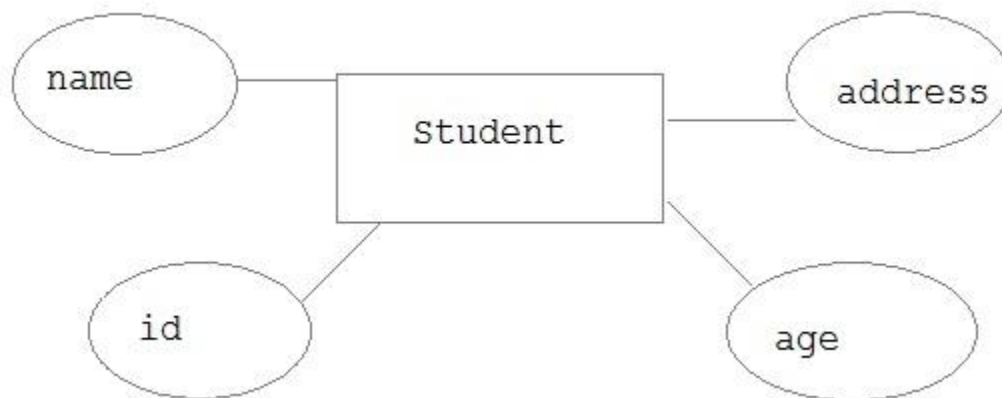
Weak Entity: Weak entity is an entity that depends on another entity. Weak entity doesn't have key attribute of their own. Double rectangle represents weak entity.



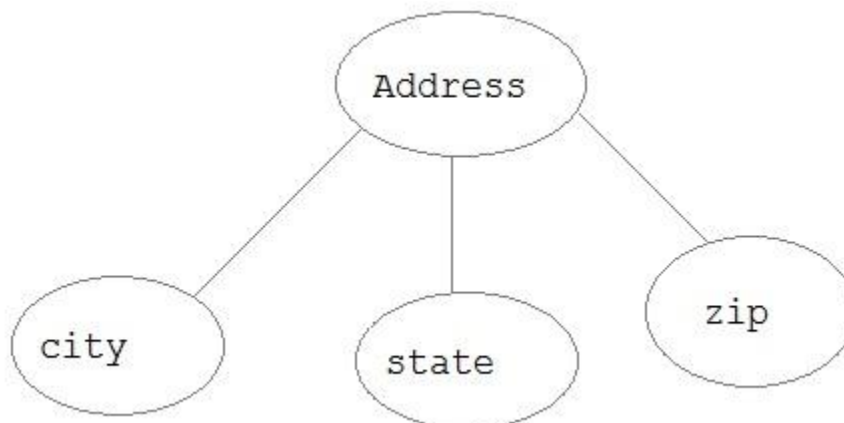
Attribute: An Attribute describes a property or characteristic of an entity. For example, Name, Age, Address etc can be attributes of a Student. An attribute is represented using eclipse.



Key Attribute: Key attribute represents the main characteristic of an Entity. It is used to represent Primary key. Ellipse with underlying lines represents Key Attribute.



Composite Attribute: An attribute can also have their own attributes. These attributes are known as Composite attribute.



3) **Relationship:** A Relationship describes relations between entities. Relationship is represented using diamonds.



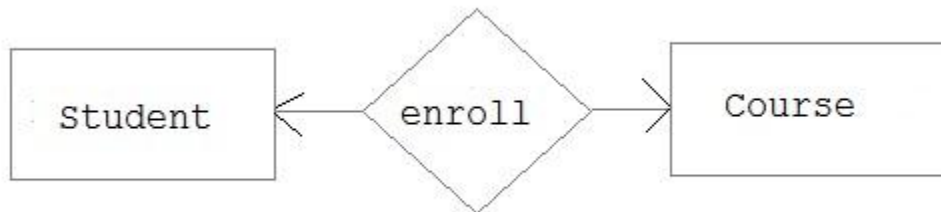
There are three types of relationship that exist between Entities.

- Binary Relationship
- Recursive Relationship
- Ternary Relationship

Binary Relationship

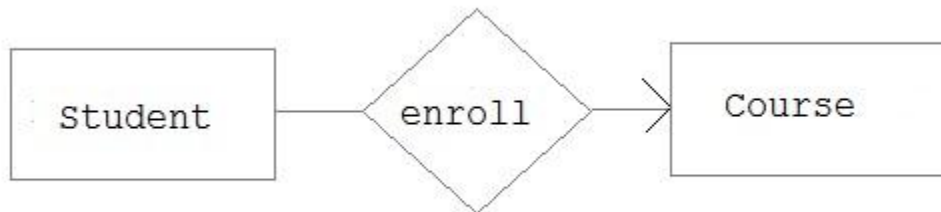
1. Binary Relationship means relation between two Entities. This is further divided into three types.

1. **One to One:** This type of relationship is rarely seen in real world.



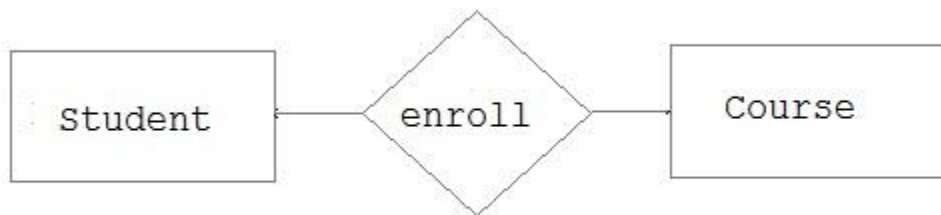
The above example describes that one student can enroll only for one course and a course will also have only one Student. This is not what you will usually see in relationship.

- **One to Many:** It reflects business rule that one entity is associated with many number of same entity. For example, Student enrolls for only one Course but a Course can have many Students.

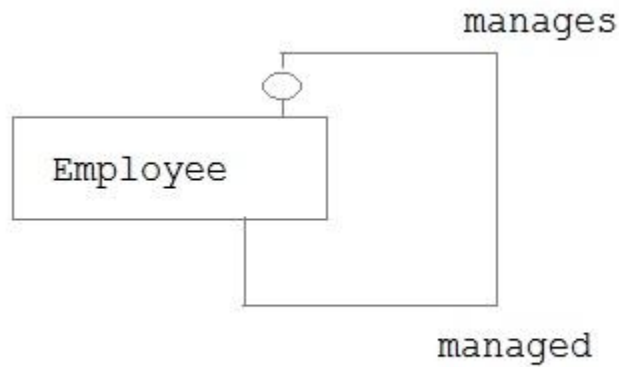


The arrows in the diagram describes that one student can enroll for only one course.

2. **Many to Many :** The below diagram shows the many to many relationship



Recursive Relationship: When an Entity is related with itself it is known as Recursive Relationship.



Ternary Relationship: Relationship of degree three is called Ternary relationship.

Generalization, Specialization and Aggregation

Generalization is a bottom-up approach in which two lower level entities combine to form a higher level entity. In generalization, the higher level entity can also combine with other lower level entity to make further higher level entity.

Specialization: Specialization is opposite to Generalization. It is a top-down approach in which one higher level entity can be broken down into two lower level entities. In specialization, some higher level entities may not have lower-level entity sets at all.

Aggregation: Aggregation is a process when relation between two entities is treated as a single entity. Here the relation between Center and Course, is acting as an Entity in relation with Visitor.

Transaction Management

Transaction Control Language (TCL) commands are used to manage transactions in database. These are used to manage the changes made by DML statements. It also allows statements to be grouped together into logical transactions.

Commit command

Commit command is used to permanently save any transaction into database.

Following is Commit command's syntax,

commit;

Rollback command

This command restores the database to last committed state. It is also use with savepoint command to jump to a savepoint in a transaction.

Following is Rollback command's syntax,

rollback to savepoint-name;

Savepoint command

savepoint command is used to temporarily save a transaction so that you can rollback to that point whenever necessary.

Following is savepoint command's syntax,

savepoint savepoint-name;

RDBMS CONCEPTS

A **Relational Database management System** (RDBMS) is a database management system based on relational model introduced by E.F Codd. In relational model, data is represented in terms of tuples (rows).

RDBMS is used to manage Relational database. **Relational database** is a collection of organized set of tables from which data can be accessed easily. Relational Database is most commonly used database. It consists of number of tables and each table has its own primary key.

What is Table?

In Relational database, a **table** is a collection of data elements organized in terms of rows and columns. A table is also considered as convenient representation of **relations**. But a table can have duplicate tuples while a true **relation** cannot have duplicate tuples. Table is the simplest form of data storage. Below is an example of Employee table.

ID	Name	Age	Salary
1	Adam	34	13000
2	Alex	28	15000
3	Stuart	20	18000
4	Ross	42	19020

What is a Record?

A single entry in a table is called a **Record** or **Row**. A **Record** in a table represents set of related data. For example, the above **Employee** table has 4 records. Following is an example of single record.

1	Adam	34	13000
---	------	----	-------

What is Field ?

A table consists of several records (row), each record can be broken into several smaller entities known as Fields. The above Employee table consists of four fields, ID, Name, Age and Salary.

What is a Column?

In Relational table, a column is a set of value of a particular type. The term Attribute is also used to represent a column. For example, in Employee table, Name is a column that represents names of employee.

Name
Adam
Alex
Stuart
Ross

Codd's Rule

E.F Codd was a Computer Scientist who invented **Relational model** for Database management. Based on relational model, **Relation database** was created. Codd proposed 13 rules popularly known as **Codd's 12 rules** to test DBMS's concept against his relational model. Codd's rule actually defines what quality a DBMS requires in order to become a Relational Database Management System (RDBMS). Till now, there is hardly any commercial product that follows all the 13 Codd's rules. Even **Oracle** follows only eight and half out (8.5) of 13. The Codd's 12 rules are as follows.

Rule zero

This rule states that for a system to qualify as an **RDBMS**, it must be able to manage database entirely through the relational capabilities.

Rule 1: Information rule

All information (including metadata) is to be represented as stored data in cells of tables. The rows and columns have to be strictly unordered.

Rule 2: Guaranteed Access

Each unique piece of data (atomic value) should be accessible by: **Table Name + primary key (Row) + Attribute (column)**.

NOTE: Ability to directly access via POINTER is a violation of this rule.

Rule 3: Systematic treatment of NULL

Null has several meanings; it can mean missing data, not applicable or no value. It should be handled consistently. Primary key must not be null. Expression on **NULL** must give null.

Rule 4: Active Online Catalog

Database dictionary (catalog) must have description of **Database**. Catalog to be governed by same rule as rest of the database. The same query language to be used on catalog as on application database.

Rule 5: Powerful language

One well defined language must be there to provide all manners of access to data. Example: **SQL**. If a file supporting table can be accessed by any manner except SQL interface, then its a violation to this rule.

Rule 6: View Updation rule

All view that is theoretically updatable should be updatable by the system.

Rule 7: Relational Level Operation

There must be Insert, Delete, and Update operations at each level of relations. Set operation like Union, Intersection and minus should also be supported.

Rule 8 : Physical Data Independence

The physical storage of data should not matter to the system. If say, some file supporting table were renamed or moved from one disk to another, it should not affect the application.

Rule 9: Logical Data Independence

If there is change in the logical structure (table structures) of the database the user view of data should not change. Say, if a table is split into two tables, a new view should give result as the join of the two tables. This rule is most difficult to satisfy.

Rule 10: Integrity Independence

The database should be able to conforce its own integrity rather than using other programs. Key and Check constraints, trigger etc should be stored in Data Dictionary. This also makes **RDBMS** independent of front-end.

Rule 11: Distribution Independence

A database should work properly regardless of its distribution across a network. This lays foundation of distributed database.

Rule 12: Nonsubversion rule

If low level access is allowed to a system it should not be able to subvert or bypass integrity rule to change data. This can be achieved by some sort of locking or encryption.

DATA COMMUNICATION AND NETWORKING

Network Architecture:

Network architecture is the design of a communications **network**. It is a framework for the specification of a **network's** physical components and their

functional organization and configuration, its operational principles and procedures, as well as data formats used in its operation.

Transmission:

There are two type of transmission:

1. **Parallel Transmission:** In this mode, message information is transmitted bit by bit over the link. The transmission speed of the transmitting site depends on the signaling speed. The **Signaling Rate** is defined for the communication device and usually expressed in Baud.
2. **Parallel Transmission:** In this Transmission mode, Each Bit is assigned a specific, separate channel number and all the bits are transmitted over the different channels.

There are different configurations of transmission of a signal for example

1. **Simplex:** Information always flows in one direction.
2. **Half Simplex:** Allows the transmission of a signal in one direction at a time.
3. **Full Duplex:** Allows the transmission of a signal in both directions simultaneously.

Multiplexing:

Multiplexing is a set of technique that allows a simultaneous transmission of multiple signals across a single data link.

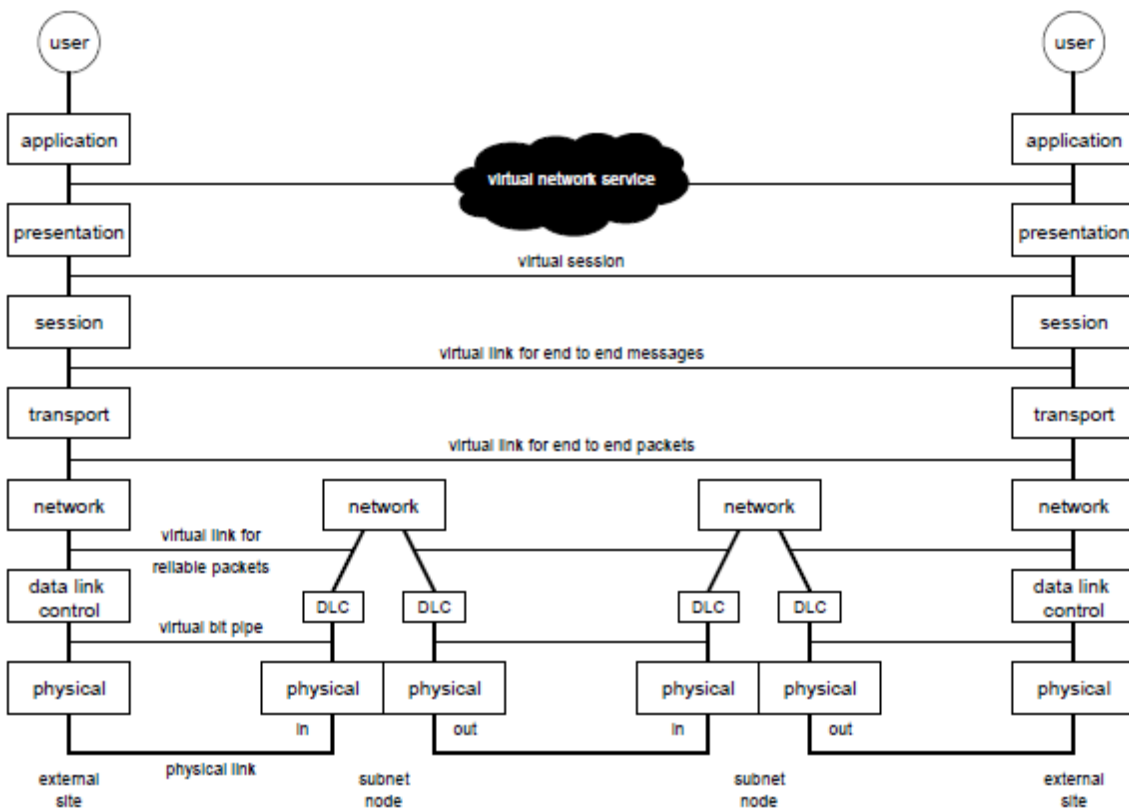
Type of Multiplexing:

1. **Frequency division multiplexing:** is an analog technique that can be applied when the bandwidth of a link is greater the combined band width.

2. **Time division multiplexing:** is a digital process that allows several connections to share the high bandwidth of signal.

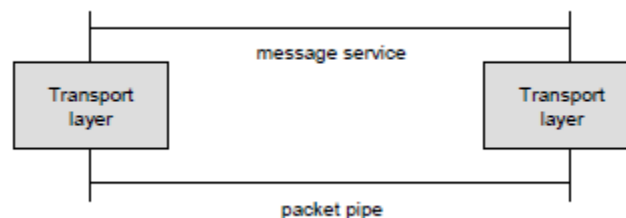
OSI MODEL

The open system interconnection Model does not define any services or protocols for OSI but instead provides a framework for coordinating the development of various standards for interconnecting different systems. It is a way of sub-dividing a communications system into smaller parts called layers. A layer is a collection of similar functions that provide services to the layer above it and receives services from the layer below it. On each layer, an instance provides services to the instances at the layer above and requests service from the layer below. The various layers of OSI Models are as follows:



Starting from Top Layer:

1. **Application layer**: This is the application that is used to access the network. Each application performs something specific to the user needs, e.g. browsing the web, transferring files, sending email, etc
2. **Presentation layer**: The main functions of the presentation layer are data formats, data encryption/decryption, data compression/decompression, etc.
3. **Session layer**: Mainly deals with access rights in setting up sessions, e.g. who has access to particular network services, billing functions, etc. There is not a strong agreement about the definition of these three top layers. Usually the focus is on the Transport layer, the Network layer, and the DLC layer.
4. **Transport layer**: While the network layer (see section below) provides end-to-end packet pipe to the transport layer, the transport layer provides end-to-end message service to the top layers.



Functions of the transport layer include:

- Breaking messages into packets and reassembling packets into messages (packets of suitable size to network) Resequencing packets at destination to retrieve correct order (e.g. Datagram)
- Achieving end-to-end reliable communication in case network is not reliable,
- Recover from errors and failures (arbitrary networks can join the Internet!)
- Flow control to prevent a fast sender from overrunning a slow receiver

- Examples of transport protocols for the Internet are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). When combined with the IP protocol at the network layer, we refer to TCP as TCP/IP.

5. **Network layer:** The main function of the network layer is to route each packet to the proper outgoing DLC or to the transport layer (if the node is the destination). Typically, the network layer adds its own header (e.g. source/destination or VC number) to the packet received from the transport layer to accomplish this routing function.

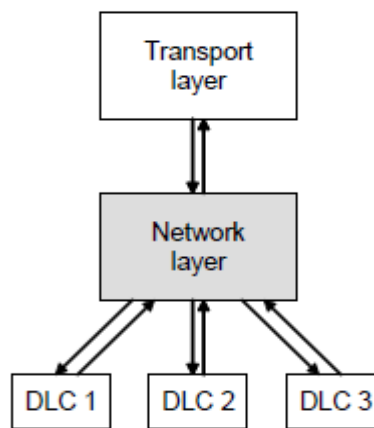


Figure 3: Routing

Headers represent a general mechanism across the layers. Each layer/protocol provides a service to the upper layer/protocol, and peer processes/protocols communicate information through the headers. The DLC layer adds also a Trailer for error detection and correction.

6. **The DLC layer:** is responsible for error-free transmission of packets over a single link. The goal is to ensure that every packet is delivered once, only once, without errors and in order. To accomplish this task, DLC adds its own header/trailer. For instance, the header may contain sequence numbers to ensure delivery of packets in order. The packet thus modified is called a *frame*.

7. **Physical layer:** is responsible for the actual transmission of bits over a link. This layer is usually the network hardware. Higher layers, like DLC, must deal with transmission errors due to noise and signal power loss. A simple model for the

physical layer is the Binary Symmetric Channel with a probability p of flipping each bit independently, i.e. $p f_0 ! 1g = p f_1 ! 0g = p$. However, in practice errors are bursty. There are a number of delays associated with the physical transmission:

>>Propagation delay: time it takes for signal to travel from one end of link to another = distance/speed of light

>>Bandwidth: number of bits that can be transmitted over a period of time, i.e. bits per second (bps)

>>Latency of packet = Propagation delay + size of packet/Bandwidth + Queuing delay

>>RTT = Round Trip Time for exchanging small messages $\frac{1}{2}$ (Propagation delay+ Queuing)

TCP/IP MODEL

A TCP/IP network is generally a heterogeneous network, meaning there are many different types of network computing devices attached.

Layering Model

In the early days of networking, before the rise of the ubiquitous Internet, the International Organization for Standardization (ISO) developed a layering model whose terminology persists today.

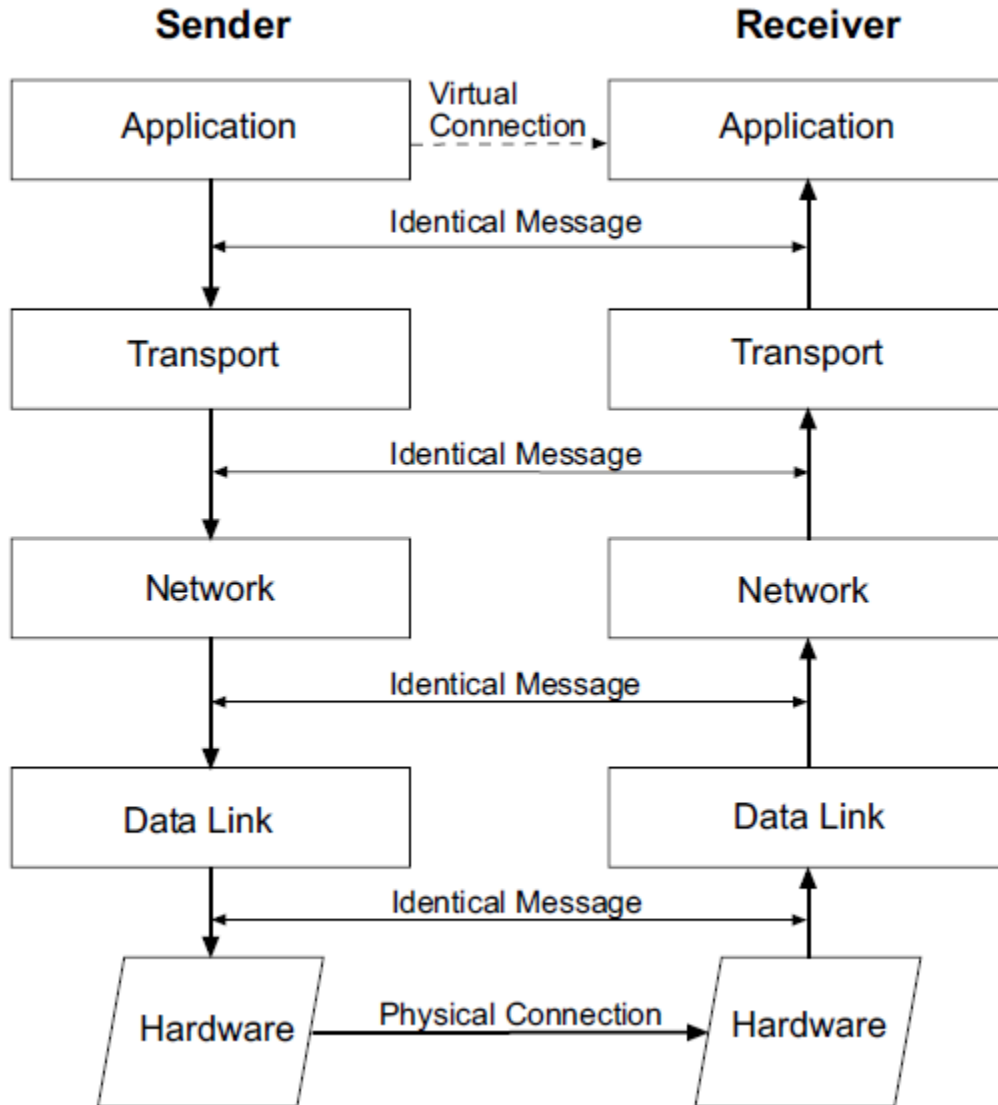
	Name of Layer	Purpose of Layer
Layer 5	Application	Specifies how a particular application uses a network.
Layer 4	Transport	Specifies how to ensure reliable transport of data.
Layer 3	Internet	Specifies packet format and routing.
Layer 2	Network	Specifies frame organization and transmittal.
Layer 1	Physical	Specifies the basic network hardware.

TCP/IP Protocol Stack

TCP/IP is the protocol suite upon which all Internet communication is based. Different vendors have developed other networking protocols, but even most network operating systems with their own protocols, such as Netware, support TCP/IP. It has become the de facto standard.

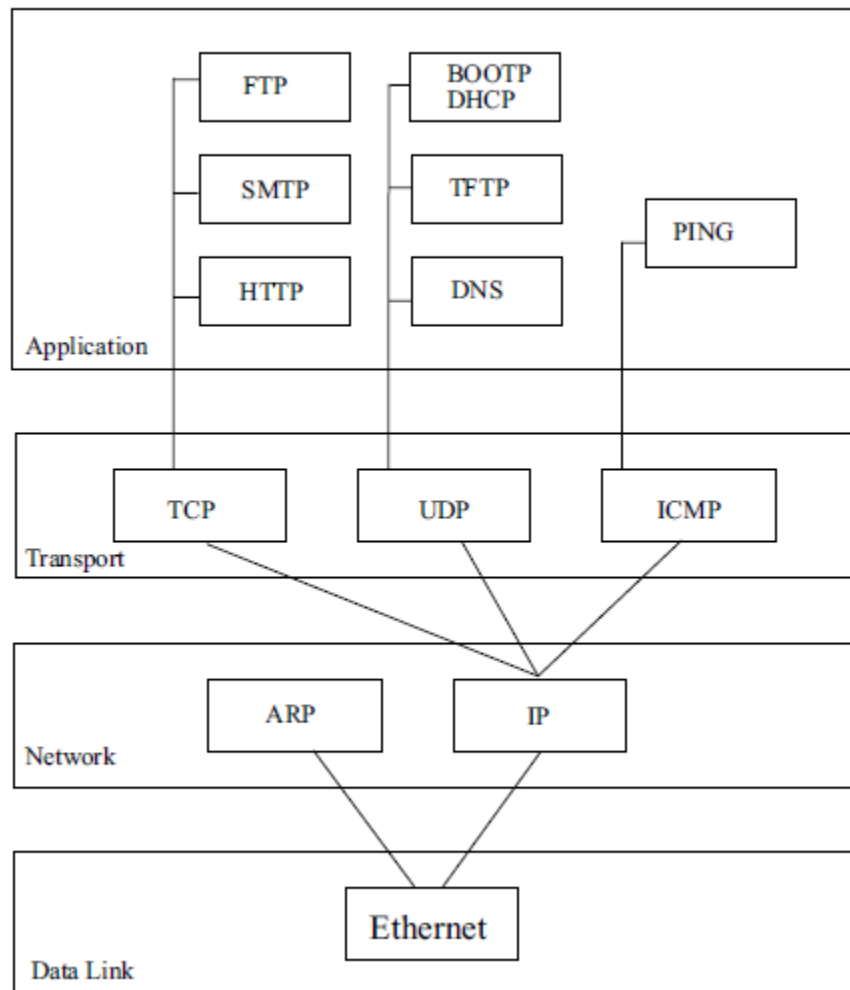
Protocols are sometimes referred to as protocol stacks or protocol suites. A protocol stack is an appropriate term because it indicates the layered approach used to design the networking software

Figure 4.1 Flow of Data Between Two Computers Using TCP/IP Stacks



TCP/IP Protocol Flow:

Figure 5.1 TCP/IP Protocol Flow



IP ADDRESSING:

IP: IP provides communication between hosts on different kinds of networks (i.e., different data-link implementations such as Ethernet and Token Ring). It is a connectionless, unreliable packet delivery service.

Connectionless means that there is no handshaking, each packet is independent of any other packet. It is unreliable because there is no guarantee that a packet gets delivered; higher-level protocols must deal with that.

IP Address

IP defines an addressing scheme that is independent of the underlying physical address (e.g.; 48-bit MAC address). IP specifies a unique 32-bit number for each host on a network. This number is known as the Internet Protocol Address, the IP Address or the Internet Address. These terms are interchangeable. Each packet sent across the internet contains the IP address of the source of the packet and the IP address of its destination. For routing efficiency, the IP address is considered in two parts: the prefix which identifies the physical network, and the suffix which identifies a computer on the network. A unique prefix is needed for each network in an internet. For the global Internet, network numbers are obtained from Internet Service Providers (ISPs) . ISPs coordinate with a central organization called the Internet Assigned Number Authority (IANA).

IP Address Classes

The first four bits of an IP address determine the class of the network. The class specifies how many of the remaining bits belong to the prefix (aka Network ID) and to the suffix (aka Host ID). The first three classes, A, B and C, are the primary network classes.

Class	First 4 Bits	Number Of Prefix Bits	Max # Of Networks	Number Of Suffix Bits	Max # Of Hosts Per Network
A	0xxx	7	128	24	16,777,216
B	10xx	14	16,384	16	65,536
C	110x	21	2,097,152	8	256
D	1110	Multicast			
E	1111	Reserved for future use.			

When interacting with mere humans, software uses dotted decimal notation; each 8 bits is treated as an unsigned binary integer separated by periods. IP reserves host address 0 to denote a network. 140.211.0.0 denotes the network that was assigned the class B prefix 140.211.

Netmasks: Netmasks are used to identify which part of the address is the Network ID and which part is the Host ID. This is done by a logical bitwise-AND of the IP address and the netmask. For class A networks the netmask is always 255.0.0.0; for class B networks it is 255.255.0.0 and for class C networks the netmask is 255.255.255.0.

Subnet Address

All hosts are required to support subnet addressing. While the IP address classes are the convention, IP addresses are typically subnetted to smaller address sets that do not match the class system. The suffix bits are divided into a subnet ID and a host ID. This makes sense for class A and B networks, since no one attaches as many hosts to these networks as is allowed. Whether to subnet and how many bits to use for the subnet ID is determined by the local network administrator of each network. If subnetting is used, then the netmask will have to reflect this fact. On a class B network with subnetting, the netmask would not be 255.255.0.0. The bits of the Host ID that were used for the subnet would need to be set in the netmask.

Directed Broadcast Address

IP defines a directed broadcast address for each physical network as all ones in the host ID part of the address. The network ID and the subnet ID must be valid network and subnet values. When a packet is sent to a network's broadcast address, a single copy travels to the network, and then the packet is sent to every host on that network or sub network.

Limited Broadcast Address

If the IP address is all ones (255.255.255.255), this is a limited broadcast address; the packet is addressed to all hosts on the current (sub) network. A router will not forward this type of broadcast to other (sub) networks.

Some **terminology** used in Data communication and Networking:

1. **Automatic repeat request:** This error control technique provides error recovery after the error is detected in the event of error detected by the ARQ at the receiving site, the receiver requests the sending site to retransmit the protocol data unit. This provides reliable data link. There are three version of ARQ:

- A. **Stop-and-wait ARQ:** A sending station a frame to the destination station and waits until it receives an acknowledgement from the destination station.
- B. **Go-Back-N ARQ:** If an error is detected in any frame (when an acknowledgment arrives at sending station) or the acknowledgment is lost or it is times out, in all three cases, the sending station will retransmit the same frame until it is received error-free on the receiving side.
- C. **Go-Back-Select ARQ:** this is similar to Go-Back-N ARQ, the only difference is being that here the frames behind the error frame are stored in a buffer at the receiving site until the error frame received error-free.

2. Datagram: It is a basic transfer unit associated with a packet-switched network in which the delivery, arrival time, and order of arrival are not guaranteed by the network service.

3. Modulation: It is an operation which translates a modulating signal into another signal using a constant carrier signal of high frequency. The main classes of modulation techniques:

A. **Analog Modulation (AM)** is of three type: Amplitude, Frequency and Phase Modulation

B. **Digital Modulation (DM)**

4. Integrity: means the data must arrive at the receiver exactly as it was send. There must b no change in the data. Technique used for the data integrity:

A. **Parity Check;** which is of 4 types:

1. Simple Parity Check: A redundant bit called the parity bit is added to every data unit so that total no of 1s in the unit becomes even.

2. Even Parity Concept: Before Transmitting the data, we pass the unit through a parity generator. The Parity generator counts no of 1s and appends the parity bit to the end. If in the end the no of parity bits are even then the whole unit is rejected.

3. **Cyclic redundancy Check (CRC):** The receiver side calculates the proper parity character from the received block of data. The calculated parity is compared with that sent in the character check field. If both are same then there is no error and if they are not same then the negative acknowledgment is sent indicating the occurrence of error.

4. **Check Sum:** If parity bit leaves some error undetected then parity bits of all the characters in the frame can incorporate an additional check for error detection. It can be implemented in many ways. eFor example 1s complement of a number.

5. **Hamming Code:** These codes were originally designed with $D_{min}=3$ which means they can detect upto two errors and correct a single error.

Error Control Methods: are used to reduce the effect of noise on the signals.

- **Block Codes:** consist of information bit, redundant bits and various Code are implemented to detect and correct the errors.
- **Convolution Codes:** They also generate codes words and they depend upon current as well as previous value of information.

6. **Switches:** A switched network consist of a series of interconnection nodes called switches. They are the devices capable of creating temporary connections between two or more devices linked to a switch.

- **Circuit Switching:** A dedicated connection is established for the duration of message between two nodes .This type of switching is done in telephone networks and some of the upcoming switched networks.
- **Packet Switching:** Data is sent in a sequence of small chunks called the packets. Each packet passed through node to node along some path leading from source to destination.

7. **Frame Relay:** Frame relay is a Virtual circuit wide area network .It operates at a high speed (1.54 - 44.376 MBPS)

- Operates in physical and data link layer.
- Allows bursty data

- It allows frame size of 9000 bytes and less expensive than other WANs.
- Has a error detection at the Data link layer only.

8. **LAN (Local area network):** are privately owned networks within a single building.

9. **MAN (Metropolitan Area Network)** covers a city. Best example of this is cable networks and MTNL.

10. **WAN (Wide Area Network)** spans a large geographical are often countries or continents.

11. **Repeater:** It is a device that works in a physical layer. Signals that carry information within a network can travel a fixed distance before attenuation endangers the integrity of data. A repeater receives the data signal and before it becomes too weak regenerates the original bit pattern.

12. **Router:** It provides interconnection between two networks. It is a networking unit which is compatible with the lower three layers.

13. A **gateway** is a network point that acts as an entrance to another network. On the Internet, a node or stopping point can be either a gateway node or a host (end-point) node. Both the computers of Internet users and the computers that serve pages to users are host nodes. The computers that control traffic within your company's network or at your local Internet service provider (ISP) are gateway nodes.

14. A **network bridge** is a network device that connects multiple network segments. It is a physical layer device. It regenerates the signal it receives. It can check the MAC addresses contained in the frame.

Operating System

Process:

A process is defined as an entity which represents the basic unit of work to be implemented in the system.

Components of a process are following:

1. Object Program: Code to be executed.
2. Data: Data to be used for executing the program.
3. Resources: While executing the program, it may require some resources.
4. Status: Verifies the status of the process execution. A process can run to completion only when all requested resources have been allocated to the process. Two or more processes could be executing the same program, each using their own data and resources

Process States:

As a **process** executes, it changes state. The state of a process is defined as the current activity of the process. State and Description of the process:

1. New :The Process is being created.
2. Ready: The process is waiting to be assigned to a processor. Ready processes are waiting to have the processor allocated to them by the operating system so that they can run.
3. Running: Process instructions are being executed (i.e. the process that is currently being executed).
4. Waiting: The process is waiting for some event to occur (such as the completion of an I/O operation).
5. Terminated: The Process has finished Execution.

Process Control Block

Each process is represented in the operating system by a process control block (PCB) also called a task control block. PCB is the data structure used by the operating system. Operating system groups all information that needs about particular process.

1. **Pointer:** Pointer points to another process control block. Pointer is used for maintaining the scheduling list.
2. **Process State:** Process state may be new, ready, running, waiting and so on.

3. Program Counter: Program Counter indicates the address of the next instruction to be executed for this process.

4. CPU registers: CPU registers include general purpose register, stack pointers, index registers and accumulator's etc. number of register and type of register totally depends upon the computer architecture.

5. Memory management information: This information may include the value of base and limit registers, the page tables, or the segment tables depending on the memory system used by the operating system. This information is useful for deallocating the memory when the process terminates.

6. Accounting Information: This information includes the amount of CPU and real time used, time limits, job or process numbers, account numbers etc.

THREAD

A Thread is a flow of execution through the process code, with its own program counter, system registers and stack. A thread is also called a light weight process. Threads provide a way to improve application performance through parallelism. Threads represent a software approach to improving performance of operating system by reducing the overhead thread is equivalent to a classical process.

Advantages of Thread :

- Thread minimizes context switching time.
- Use of threads provides concurrency within a process.
- Efficient communication.
- Economy- It is more economical to create and context switch threads.

Types Of Thread :

1. User level Thread: User Managed Threads.

2. Kernel level Threads: Operating System Managed Threads acting on kernels which is a core of an operating system.

Difference Between User and Kernel Level Threads :

S.N.	User Level Threads	Kernel Level Thread
1	User level threads are faster to create and manage.	Kernel level threads are slower to create and manage.
2	Implementation is by a thread library at the user level.	Operating system supports creation of Kernel threads.
3	User level thread is generic and can run on any operating system.	Kernel level thread is specific to the operating system.

Types Of Operating System

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

The problems with Batch Systems are following:

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

2.Time-sharing operating systems: Time sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time-sharing or multitasking is a logical extension of multiprogramming. Processor's time which is shared among multiple users simultaneously is termed as time-sharing. The main difference between Multiprogrammed Batch Systems and Time-Sharing Systems is that in case of

multiprogrammed batch systems, objective is to maximize processor use, whereas in Time-Sharing Systems objective is to minimize response time.

Advantages of Timesharing operating systems are following :

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

Disadvantages of Timesharing operating systems are following:

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

3.Distributed operating System: Distributed systems use multiple central processors to serve multiple real time application and multiple users. Data processing jobs are distributed among the processors accordingly to which one can perform each job most efficiently.

The advantages of distributed systems are following:

- With resource sharing facility user at one site may be able to use the resources available at another.
- Speedup the exchange of data with one another via electronic mail.
- If one site fails in a distributed system, the remaining sites can potentially continue operating.
- Better service to the customers.
- Reduction of the load on the host computer.
- Reduction of delays in data processing.

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

Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD.

The advantages of network operating systems are :

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

The disadvantages of network operating systems are :

- High cost of buying and running a server.
- Dependency on a central location for most operations.
- Regular maintenance and updates are required.

5.Real Time operating System: Real time system is defines as a data processing system in which the time interval required to process and respond to inputs is so small that it controls the environment. Real time processing is always on line whereas on line system need not be real time. The time taken by the system to respond to an input and display of required updated information is termed as response time. So in this method response time is very less as compared to the online processing. Real Time Operating System examples are **Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, and home-appliance controllers, Air traffic control system etc.** There are two types of real-time operating systems.

Hard real-time systems :

Hard real-time systems guarantee that critical tasks complete on time. In hard real-time systems secondary storage is limited or missing with data stored in ROM. In these systems virtual memory is almost never found.

Soft real-time systems :

Soft real time systems are less restrictive. Critical real-time task gets priority over other tasks and retains the priority until it completes. Soft real-time systems have limited utility than hard real-time systems. For example, Multimedia, virtual reality, Advanced Scientific Projects like undersea exploration and planetary rovers etc.

SEMAPHORES

Semaphore is a synchronization tool. semaphore is a value that indicates the status of common resources. A **semaphore**, in its most basic form, is a protected integer variable that can facilitate and restrict access to shared resources in a multi-processing environment. The two most common kinds of semaphores are **counting semaphores** and **binary semaphores**.

Counting semaphores represent multiple resources, while binary semaphores, as the name implies, represents two possible states (generally 0 or 1; locked or unlocked). Semaphores were invented by the late Edsger Dijkstra.

Points to Remember about:

- A semaphore can only be accessed using the following operations: **wait()** and **signal()**
- **wait()** is called when a process wants access to a resource.
- **signal()** is called when a process is done using a resource, or when the patron is finished with his meal.
- If there is only one count of a resource, a **binary semaphore** is used which can only have the values of 0 or 1. They are often used as **mutex locks**

Scheduling

The process **scheduling** is the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process on the basis of a particular strategy.

Process scheduling is an essential part of a Multiprogramming operating system. Such operating systems allow more than one process to be loaded into the executable memory at a time and loaded process shares the CPU using time multiplexing. Schedulers are of three types

- Long Term Scheduler
- Short Term Scheduler
- Medium Term Scheduler

Long Term Scheduler :

It is also called job scheduler. Long term scheduler determines which programs are admitted to the system for processing. Job scheduler selects processes from the queue and loads them into memory for execution. Process loads into the memory for CPU scheduling. The primary objective of the job scheduler is to provide a balanced mix of jobs, such as I/O bound and processor bound. It also controls the degree of multiprogramming.

Short Term Scheduler :

It is also called CPU scheduler. Main objective is increasing system performance in accordance with the chosen set of criteria. It is the change of ready state to running state of the process. CPU scheduler selects process among the processes that are ready to execute and allocates CPU to one of them. It is also known as a **Dispatcher**.

Medium Term Scheduler:

Medium term scheduling is part of the swapping. It removes the processes from the memory. It reduces the degree of multiprogramming. The medium term scheduler is in-charge of handling the swapped out-processes.

Comparison between Schedulers:

S.N.	Long Term Scheduler	Short Term Scheduler	Medium Term Scheduler
1	It is a job scheduler	It is a CPU scheduler	It is a process swapping scheduler.
2	Speed is lesser than short term scheduler	Speed is fastest among other two	Speed is in between both short and long term scheduler.
3	It controls the degree of multiprogramming	It provides lesser control over degree of multiprogramming	It reduces the degree of multiprogramming.
4	It is almost absent or minimal in time sharing system	It is also minimal in time sharing system	It is a part of Time sharing systems.

Four Major Type of Scheduling Algorithms are :

1.First Come First Serve (FCFS):

- Jobs are executed on first come, first serve basis.
- Easy to understand and implement.
- Poor in performance as average wait time is high.

2.Shortest Job First (SJF):

- Best approach to minimize waiting time.
- Impossible to implement
- Processer should know in advance how much time process will take.

3.Priority Based Scheduling :

- Each process is assigned a priority. Process with highest priority is to be executed first and so on.
- Processes with same priority are executed on first come first serve basis.
- Priority can be decided based on memory requirements, time requirements or any other resource requirement.

4.Round Robin Scheduling :

- Each process is provided a fix time to execute called quantum.
- Once a process is executed for given time period. Process is preempted and other process executes for given time period.
- Context switching is used to save states of preempted processes.

Deadlock

In a multiprogramming system, processes request resources. If those resources are being used by other processes then the process enters a waiting state. However, if other processes are also in a waiting state, we have deadlock.

A deadlock occurs if and only if the following four conditions hold in a system simultaneously:

1. **Mutual Exclusion:** At least one of the resources is non-sharable (that is; only a limited number of processes can use it at a time and if it is requested by a process while it is being used by another one, the requesting process has to wait until the resource is released.
2. **Hold and Wait:** There must be at least one process that is holding at least one resource and waiting for other resources that are being hold by other processes.
3. **No Preemption:** No resource can be preempted before the holding process completes its task with that resource.
4. **Circular Wait:** There exists a set of processes: {P1, P2, ..., Pn} such that P1 is waiting for a resource held by P2 P2 is waiting for a resource held by P3 ... Pn-1 is waiting for a resource held by Pn Pn is waiting for a resource held by P1.

Methods for handling deadlocks are:

- Deadlock prevention
- Deadlock avoidance
- Deadlock detection and recovery.

Deadlock Prevention :To prevent the system from deadlocks, one of the four discussed conditions that may create a deadlock should be discarded. The methods for those conditions are as follows:

1. **Mutual Exclusion:** In general, we do not have systems with all resources being sharable. Some resources like printers, processing units are non-sharable. So it is not possible to prevent deadlocks by denying mutual exclusion.
2. **Hold and Wait:** One protocol to ensure that hold-and-wait condition never occurs says each process must request and get all of its resources before it begins execution. Another protocol is "Each process can request resources only when it does not occupies any resources."
3. **No Preemption:** One protocol is "If a process that is holding some resources requests another resource and that resource cannot be allocated to it, then it must release all resources that are currently allocated to it."

Another protocol is "When a process requests some resources, if they are available, allocate them. If a resource it requested is not available, then we check whether it is being used or it is allocated to some other process waiting for other resources. If that resource is not being used, then the OS preempts it from the waiting process and allocate it to the requesting process. If that resource is used, the requesting process must wait."

4. **Circular Wait:** One protocol to ensure that the circular wait condition never holds is "Impose a linear ordering of all resource types." Then, each process can only request resources in an increasing order of priority.

Deadlock Avoidance: The algorithm in this Technique will dynamically examine the resource allocation operations to ensure that there won't be a circular wait on resources. When a process requests a resource that is already available, the system must decide whether that resource can immediately be allocated or not. The resource is immediately allocated only if it leaves the system in a safe state. A state is safe if the system can allocate resources to each process in some order avoiding a deadlock. A deadlock state is an unsafe state.

Banker's Algorithm is used for Deadlock Avoidance

Deadlock Detection: If a system has no deadlock prevention and no deadlock avoidance scheme, then it needs a deadlock detection scheme with recovery from

deadlock capability. For this, information should be kept on the allocation of resources to processes, and on outstanding allocation requests. Then, an algorithm is needed which will determine whether the system has entered a deadlock state. This algorithm must be invoked periodically.

Shoshani and Coffman Algorithm is used for Deadlock Detection.

Memory Partitioning

In operating systems, memory management is the function responsible for managing the computer's primary memory

The memory management function keeps track of the status of each memory location, either allocated or free. It determines how memory is allocated among competing processes, deciding who gets memory, when they receive it, and how much they are allowed. When memory is allocated it determines which memory locations will be assigned. It tracks when memory is freed or unallocated and updates the status.

Memory management techniques:

1. **Single contiguous allocation:** Single allocation is the simplest memory management technique. All the computer's memory, usually with the exception of a small portion reserved for the operating system, is available to the single application. MS-DOS is an example of a system which allocates memory in this way. An embedded system running a single application might also use this technique.
2. **Partitioned Allocation:** Partitioned allocation divides primary memory into multiple memory partitions, usually contiguous areas of memory. Each partition might contain all the information for a specific job or task. Memory management consists of allocating a partition to a job when it starts and unallocating it when the job ends. Partitioned allocation usually requires some hardware support to prevent the jobs from interfering with one another or with the operating system.
3. **Paged Memory Management:** Paged allocation divides the computer's primary memory into fixed-size units called page frames, and the program's virtual address space into pages of the same size. The hardware memory

management unit maps pages to frames. The physical memory can be allocated on a page basis while the address space appears contiguous.

4. **Segmented Memory management:** Segmented memory is the only memory management technique that does not provide the user's program with a 'linear and contiguous address space. Segments are areas of memory that usually correspond to a logical grouping of information such as a code procedure or a data array. Segments require hardware support in the form of a segment table which usually contains the physical address of the segment in memory, its size, and other data such as access protection bits and status (swapped in, swapped out, etc.)

Page Replacement

Page Replacement Algorithm:

Page replacement algorithms are the techniques using which Operating System decides which memory pages to swap out, write to disk when a page of memory needs to be allocated. Paging happens whenever a page fault occurs and a free page cannot be used for allocation purpose accounting to reason that pages are not available or the number of free pages is lower than required pages.

1.First In First Out (FIFO) algorithm :

- Oldest page in main memory is the one which will be selected for replacement.
- Easy to implement, keep a list, replace pages from the tail and add new pages at the head.

2.Optimal Page algorithm :

- An optimal page-replacement algorithm has the lowest page-fault rate of all algorithms. An optimal page-replacement algorithm exists, and has been called OPT or MIN.
- Replace the page that will not be used for the longest period of time . Use the time when a page is to be used.

3.Least Recently Used (LRU) algorithm :

- Page which has not been used for the longest time in main memory is the one which will be selected for replacement.
- Easy to implement, keep a list, replace pages by looking back into time.

4. Page Buffering algorithm:

- To get process start quickly, keep a pool of free frames.
- On page fault, select a page to be replaced.
- Write new page in the frame of free pool, mark the page table and restart the process.
- Now write the dirty page out of disk and place the frame holding replaced page in free pool.

5. Least frequently Used (LFU) algorithm:

- Page with the smallest count is the one which will be selected for replacement.
- This algorithm suffers from the situation in which a page is used heavily during the initial phase of a process, but then is never used again.

Most frequently Used (LFU) algorithm :

- This algorithm is based on the argument that the page with the smallest count was probably just brought in and has yet to be used.

SOME OTHER IMPORTANT TERMS USED IN OPERATING SYSTEM ARE:

1. **Demand Paging:** In virtual memory systems, demand paging is a type of swapping in which pages of data are not copied from disk to RAM until they are needed. In contrast, some virtual memory systems use anticipatory paging, in which the operating system attempts to anticipate which data will be needed next and copies it to RAM before it is actually required.
2. **Virtual Memory:** Virtual memory is a technique that allows the execution of processes which are not completely available in memory. The main visible advantage of this scheme is that programs can be larger than physical memory. Virtual memory is the separation of user logical memory from physical memory. This separation allows an extremely large virtual memory to be provided for programmers when only a smaller physical memory is available.
3. **Daisy Chain:** When device A has a cable that plugs into device B, and device B has a cable that plugs into device C, and device C plugs into a port on the computer, this arrangement is called a daisy chain. It usually operates as a bus.

4. **Polling**: Polling is a process by which a host waits for controller response. It is a looping process, reading the status register over and over until the busy bit of status register becomes clear. The controller uses/sets the busy bit when it is busy working on a command, and clears the busy bit when it is ready to accept the next command.
5. **Direct Memory Access**: Many computers avoid burdening the main CPU with programmed I/O by offloading some of this work to a special purpose processor. This type of processor is called, a Direct Memory Access (DMA) controller. A special control unit is used to transfer block of data directly between an external device and the main memory, without intervention by the processor. This approach is called Direct Memory Access (DMA).
6. **Interrupts**: The CPU hardware uses an interrupt request line wire which helps CPU to sense after executing every instruction.
7. **File structure** File structure is a structure, which is according to a required format that operating system can understand.
8. **Paging**: External fragmentation is avoided by using paging technique. Paging is a technique in which physical memory is broken into blocks of the same size called pages (size is power of 2, between 512 bytes and 8192 bytes). When a process is to be executed, it's corresponding pages are loaded into any available memory frames.
9. **Segmentation**: Segmentation is a technique to break memory into logical pieces where each piece represents a group of related information. For example, data segments or code segment for each process, data segment for operating system and so on. Segmentation can be implemented using or without using paging.

NETWORK SECURITY

Network Security comprises of topics such as Cyber crimes, Risk Management, Firewall and cryptography.

Cyber Crimes: Computer crime is aimed at stealing the computer, damaging information or stealing information. The Use of computer to carry out any conventional criminal act such as fraud is called Cyber Crime.

Common Hacking Methods:

- **Sniffing:** The term sniffing refers to finding a user's password. There are three ways to sniff password: Password sharing, Password guessing and password capture. Password share sharing is the most common of three.
- **Social Engineering** is the act of breaking corporate security by manipulating employees into divulging confidential information. It uses psychological tricks to gain trust, rather than technical cracking techniques. Social Engineering includes scams such as obtaining a password by pretending to be an employee, leveraging social media to identify new employees more easily tricked into providing customer information, and any other attempt to breach security by gaining trust. Social engineering attacks motivated primarily by financial gain. This method is also called **Phishing**
- **Spoofing:** Spoofing is the action of making something look like something that it is not in order to gain unauthorized access to a user's private information. The idea of spoofing originated in the 1980s with the discovery of a security hole in the TCP protocol. Today spoofing exists in various forms namely IP, URL and Email spoofing. IP Spoofing is of two types:
- **Man in the Middle Attack:** In a Man-in-the-Middle attack, the message sent to a recipient is intercepted by a third-party which manipulates the packets and resends its own message.
- **Denial Of Service Attack:** A DoS attack is when an attacker floods a system with more packets than its resources can handle. This then causes the system to overload and shut down. The source address is spoofed making it difficult to track from where the attacks are taking place.
- **URL spoofing** is sometimes used to direct a user to a fraudulent site and by giving the site the same look and feel as the original site the user attempts to login with a username and password. The hacker collects the username and password then displays a password error and directs the user to the legitimate site. Using this technique the hacker could create a series of fake websites and steal a user's private information unknowingly.

- **Email spoofing** is the act of altering the header of an email so that the email appears to be sent from someone else.

RISK MANAGEMENT

Risk Analysis Terminology

Asset - Anything with value and in need of protection.

Threat - An action or potential action with the propensity to cause damage.

Vulnerability - A condition of weakness. If there were no vulnerabilities, there would be no concern for threat activity.

Countermeasure - Any device or action with the ability to reduce vulnerability.

Expected Loss - The anticipated negative impact to assets due to threat manifestation.

Impact - Losses as a result of threat activity are normally expressed in one or more impact areas. Four areas are commonly used; Destruction, Denial of Service, Disclosure, and Modification.

Data Encryption Techniques

Encryption is a technique for transforming information on a computer in such a way that it becomes unreadable. So, even if someone is able to gain access to a computer with personal data on it, they likely won't be able to do anything with the data unless they have complicated, expensive software or the original data key.

The basic function of encryption is essentially to translate normal text into ciphertext. Encryption can help ensure that data doesn't get read by the wrong people, but can also ensure that data isn't altered in transit, and verify the identity of the sender.

3 different encryption methods

There are three different basic encryption methods, each with their own advantages :

- **Hashing**

Hashing creates a unique, fixed-length signature for a message or data set.

Each “hash” is unique to a specific message, so minor changes to that message would be easy to track. Once data is encrypted using hashing, it cannot be reversed or deciphered. Hashing, then, though not technically an encryption method as such, is still useful for proving data hasn’t been tampered with.

- **Symmetric methods**

Symmetric encryption is also known as private-key cryptography, and is called so because the key used to encrypt and decrypt the message must remain secure, because anyone with access to it can decrypt the data. Using this method, a sender encrypts the data with one key, sends the data (the ciphertext) and then the receiver uses the key to decrypt the data.

- **Asymmetric methods**

Asymmetric encryption, or public-key cryptography, is different than the previous method because it uses two keys for encryption or decryption (it has the potential to be more secure as such). With this method, a public key is freely available to everyone and is used to encrypt messages, and a different, private key is used by the recipient to decrypt messages.

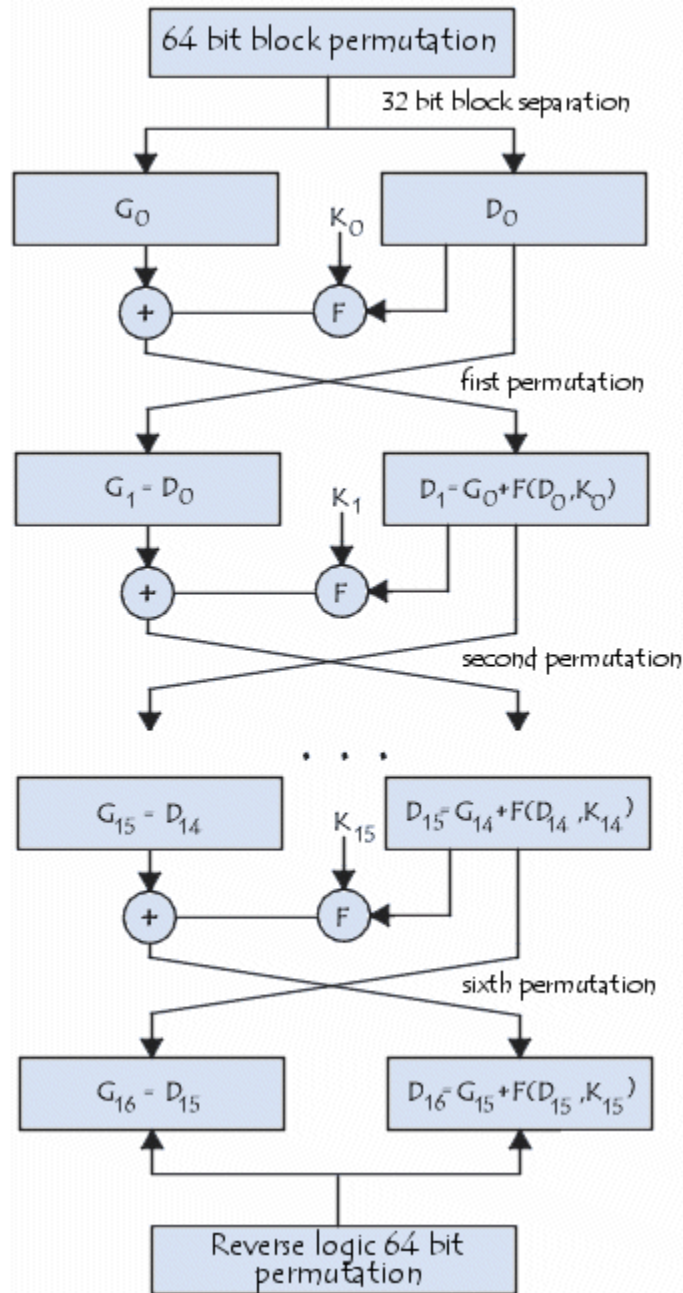
DES

DES works by using the same key to encrypt and decrypt a message, so both the sender and the receiver must know and use the same private key. Once the go-to, symmetric-key algorithm for the encryption of electronic data, DES has been superseded by the more secure Advanced Encryption Standard (AES) algorithm.

DES uses a 64-bit key, but eight of those bits are used for parity checks, effectively limiting the key to 56-bits.

The main parts of the algorithm are as follows:

- Fractioning of the text into 64-bit (8 octet) blocks;
- Initial permutation of blocks;
- Breakdown of the blocks into two parts: left and right, named *L* and *R*;
- Permutation and substitution steps repeated 16 times (called **rounds**);
- Re-joining of the left and right parts then inverse initial permutation.



The RSA Algorithm

RSA encrypts messages through the following algorithm, which is divided into 3 steps. Those three steps are Key Generation, Encryption and Decryption.

A binary Plain text is divided into blocks and a block is represented by an integer between 0 and $n-1$. This representation is necessary because the

RSA encrypts integers. The encryption key is a pair where e is a positive integer. The message block M is encrypted by raising it to the e th power modulo n i.e the cypher text corresponding to the message M is given by $C = M \text{ modulo } n$

IP Security

IP Security, a set of protocols developed by the IETF to support secure exchange of packets at the IP layer. IPsec has been deployed widely to implement Virtual Private Networks (VPNs).

IPsec supports two encryption modes: Transport and Tunnel. Transport mode encrypts only the data portion (payload) of each packet, but leaves the header untouched. The more secure Tunnel mode encrypts both the header and the payload. On the receiving side, an IPsec-compliant device decrypts each packet.

For IPsec to work, the sending and receiving devices must share a public key. This is accomplished through a protocol known as Internet Security Association and Key Management Protocol/Oakley (ISAKMP/Oakley), which allows the receiver to obtain a public key and authenticate the sender using digital certificates.

IPSec

IPSec is not a protocol, but a set of services provides various types of protection such as:

- Encryption of user data for privacy
- Authentication of the integrity of a message
- Protection for various types of attack such as replay attack
- Ability to negotiate key and security algorithms
- Two security modes: Tunnel and Transport

IPSec General Operation

Devices to work using IPSec must:

- They must agree on a set of security protocols to use, so that each one sends data in a format the other can understand.
- They must decide on a specific encryption algorithm

They must exchange keys that are used to “unlock” data that has been cryptographically encoded.

FIREWALL

A firewall is a system or group of systems that enforces an access control policy between two or more networks. The actual means by which this is accomplished varies widely, but in principle, the firewall can be thought of as a pair of mechanisms: one which exists to block traffic, and the other which exists to permit traffic.

- Generally, firewalls are configured to protect against unauthenticated interactive logins from the “outside” world. This, more than anything, helps prevent vandals from logging into machines on your network.
- Firewalls can’t protect against tunneling over most application protocols to trojaned or poorly written clients. Tunneling “bad” things over HTTP, SMTP, and other protocols is quite simple and trivially demonstrated
- Basically there are three types of firewalls: Network Layer, Application Layer and Hybrid Firewalls.

Important Security Terminologies

- **Attack** In the context of computer/network security, an attack is an attempt to access resources on a computer or a network without authorization, or to bypass security measures that are in place.
- **Audit** To track security-related events, such as logging onto the system or network, accessing objects, or exercising user/group rights or privileges. Availability of data Reliable and timely access to data.
- **Breach** Successfully defeating security measures to gain access to data or resources without authorization, or to make data or resources available to unauthorized persons, or to delete or alter computer files.
- **Brute force attack** Attempt to “crack” passwords by sequentially trying all possible combinations of characters until the right combination works to allow access. Buffer a holding area for data.
- **Buffer overflow:** A way to crash a system by putting more data into a buffer than the buffer is able to hold.

- **Countermeasures** Steps taken to prevent or respond to an attack or malicious code.
- **Cracker** A hacker who specializes in “cracking” or discovering system passwords to gain access to computer systems without authorization. See also hacker.
- **Crash** Sudden failure of a computer system, rendering it unusable.
- **Exposure** A measure of the extent to which a network or individual computer is open to attack, based on its particular vulnerabilities, how well known it is to hackers, and the time duration during which intruders have the opportunity to attack.
- **Hacker** A person who spends time learning the details of computer programming and operating systems, how to test the limits of their capabilities, and where their vulnerabilities lie.
- **Malicious code** A computer program or script that performs an action that intentionally damages a system or data, that performs another unauthorized purpose, or that provides unauthorized access to the system.
- **Risk management** The process of identifying, controlling, and either minimizing or completely eliminating events that pose a threat to system reliability, data integrity, and data confidentiality
- **Threat** A potential danger to data or systems. A threat agent can be a virus; a hacker; a natural phenomenon, such as a tornado; a disgruntled employee; a competitor, and other menaces.
- **Trojan horse** A computer program that appears to perform a desirable function but contains hidden code that is intended to allow unauthorized collection, modification or destruction of data.
- **Virus** A program that is introduced onto a system or network for the purpose of performing an unauthorized action (which can vary from popping up a harmless message to destroying all data on the hard disk).
- **Worm** A program that replicates itself, spreading from one machine to another across a network.

DATA STRUCTURE

Data Structures

Data structure is a particular way of organizing **Data** in a computer so that it can be used efficiently. Different kinds of **Data structures** are suited to different kinds of applications, and some are highly specialized to specific tasks.

Linked Lists:

Linked lists can be thought of from a high level perspective as being a series of nodes. Each node has at least a single pointer to the next node, and in the last node's case a null pointer representing that there are no more nodes in the linked list.

linked lists in DSA have the following characteristics:

1. Insertion is $O(1)$
2. Deletion is $O(n)$
3. Searching is $O(n)$

This data structure is trivial, but linked lists have a few key points which at times make them very attractive:

1. the list is dynamically resized, thus it incurs no copy penalty like an array or vector would eventually incur; and
2. insertion is $O(1)$.

Singly Linked List

Singly linked lists is a self referential data structure. A list of elements, with a head and a tail; each element points to another of its own kind.

Double Linked List is a self referential data structure. A list of elements, with a head and a tail; each element points to another of its own kind in front of it, as well as another of its own kind, which happens to be behind it in the sequence.

Circular Linked List: Linked list with no head and tail - elements point to each other in a circular fashion.

Binary Search Tree

a **binary search tree (BST)**, sometimes also called an **ordered** or **sorted binary tree**, is a node-based binary tree data structure where each node has a comparable key (and an associated value) and satisfies the restriction that the key in any node is larger than the keys in all nodes in that node's left sub-tree and smaller than the keys in all nodes in that node's right sub-tree. Each node has no more than two child nodes. Each child must either be a leaf node or the root of another binary search tree. The left sub-tree contains only nodes with keys less than the parent node; the right sub-tree contains only nodes with keys greater than the parent node. BSTs are also dynamic data structures, and the size of a BST is only limited by the amount of free memory in the operating system.

Advantages of BST:

- Binary Search Tree is fast in insertion and deletion etc. when balanced.
- Very efficient and its code is easier than other data structures.
- Stores keys in the nodes in a way that searching, insertion and deletion can be done efficiently

Disadvantages of BST:

- The shape of the binary search tree totally depends on the order of insertions, and it can be degenerated.
- When inserting or searching for an element in binary search tree, the key of each visited node has to be compared with the key of the element to be inserted or found, i.e., it takes a long time to search an element in a binary search tree.

Heap

In computer science, a heap is a specialized tree-based data structure that satisfies the heap property: If A is a parent node of B then the key of node A is ordered with respect to the key of node B with the same ordering applying across the heap. Heaps can then be classified further as either "max heap" and "min heap. Heaps are crucial in several efficient graph algorithms such as Dijkstra's algorithm, and in the sorting algorithm heapsort.

Queues

Queues are an essential data structure that are found in vast amounts of software from user mode to kernel mode applications that are core to the system.

Fundamentally they honour a First in First out (FIFO) strategy, that is the item first put into the queue will be the first served, the second item added to the queue will be the second to be served and so on.

Historically queues always have the following three core methods:

Enqueue: places an item at the back of the queue;

Dequeue: retrieves the item at the front of the queue, and removes it from the queue;

Peek: 1 retrieves the item at the front of the queue without removing it from the queue.

Queues can be ever so useful; for example the Windows CPU scheduler uses a different queue for each priority of process to determine which should be the next process to utilise the CPU for a specified time quantum. Normal queues have constant insertion and deletion run times.

A standard queue:

The main property of a queue is that we have access to the item at the front of the queue. The queue data structure can be efficiently implemented using a singly linked list

Priority Queue

Items in a priority queue being ordered by priority it remains the same as a normal queue, you can only access the item at the front of the queue.

Double Ended Queue

Double ended queue allows you to access the items at both the front, and back of the queue. A double ended queue is commonly known as a deque.

STACKS

A stack is a linear data structure for collection of items , with the restriction that item can be added one at a time and can only be removed in the reverse order in which they were added. The last item represents the top of the stack. Such a stack resembles a stack

of trays in a cafeteria, or stack of boxes. Only the top tray can be removed from the stack and it is the last one that was added to the stack.

A stack is defined in terms of its behavior. The common operations associated with a

stack are as follows:

1. **push**: adds a new item on top of a stack.
2. **pop**: removes the item on the top of a stack
3. **isEmpty**: Check to see if the stack is empty
4. **isFull**: Check to see if stack is already full
5. **returnTop**: Indicate which item is at the top

A stack is a dynamic structure. It changes as elements are added to and removed from it. It is also known as a LIFO (Last In First Out) structure.

BASIC CONCEPT OF OBJECT ORIENTED PROGRAMMING LANGUAGE:

Object

Any entity that has state and behavior is known as an object. For example: chair, pen, table, keyboard, bike etc. It can be physical and logical.

Class

Collection of objects is called class. It is a logical entity.

Inheritance

When one object acquires all the properties and behaviors of parent object i.e. known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism.

Polymorphism

When one task is performed by different ways i.e. known as polymorphism. For example: to converse the customer differently, to draw something e.g. shape or rectangle etc.

In java, we use method overloading and method overriding to achieve polymorphism.

Another example can be to speak something e.g. cat speaks meow, dog barks woof etc.

Abstraction

Hiding internal details and showing functionality is known as abstraction. For example: phone call, we don't know the internal processing.

Encapsulation

Binding (or wrapping) code and data together into a single unit is known as encapsulation

Advantage of OOPs over Procedure-oriented programming language

- 1) OOPs makes development and maintenance easier where as in Procedure-oriented programming language it is not easy to manage if code grows as project size grows.
- 2) OOPs provide data hiding whereas in Procedure-oriented programming

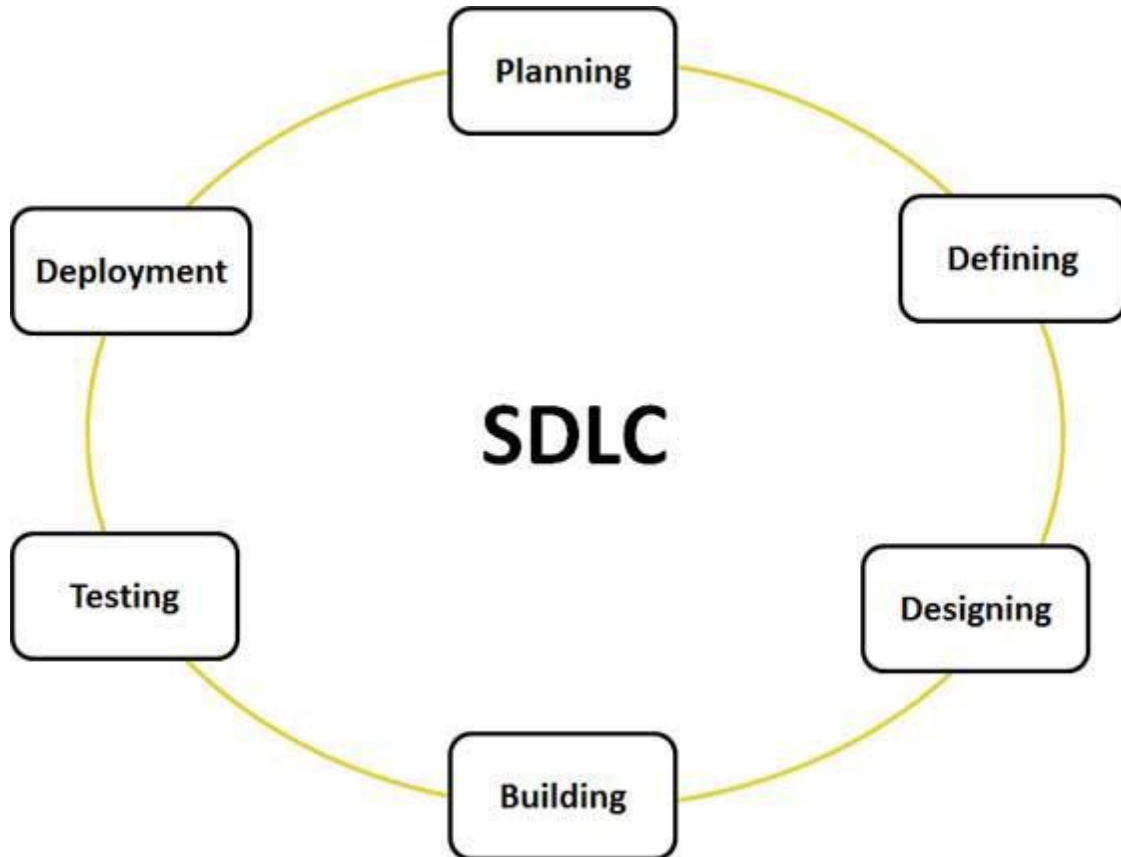
language a global data can be accessed from anywhere.

3) OOPs provides ability to simulate real-world event much more effectively. We can provide the solution of real word problem if we are using the Object-Oriented Programming language.

SOFTWARE ENGINEERING

SDLC is the acronym of Software Development Life Cycle. It is also called as Software development process. The software development life cycle (SDLC) is a framework defining tasks performed at each step in the software development process. ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.



Stage 1: *Planning and Requirement Analysis:* Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry.

Stage 2: *Defining Requirements:* Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts

Stage 3: *Designing the product architecture:* SRS is the reference for product architects to come out with the best architecture for the product to be developed

Stage 4: *Building or Developing the Product :* In this stage of SDLC the actual development starts and the product is built

Stage 5: *Testing the Product :* This stage is usually a subset of all the stages as in the modern SDLC models

Stage 6: *Deployment in the Market and Maintenance :* Once the product is tested and ready to be deployed it is released formally in the appropriate market

Popular SDLC models followed in the industry:

- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Big Bang Model

Software engineering is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures. The outcome of software engineering is an efficient and reliable software product. The process of developing a software product using software engineering principles and methods is referred to as **Software Evolution**

Characteristics of good software

A software product can be judged by what it offers and how well it can be used. This software must satisfy on the following grounds:

- Operational
- Transitional
- Maintenance

Well-engineered and crafted software is expected to have the following characteristics:

Operational

This tells us how well the software works in operations. It can be measured on:

- Budget
- Usability
- Efficiency
- Correctness
- Functionality
- Dependability
- Security
- Safety

Transitional

This aspect is important when the software is moved from one platform to another:

- Portability
- Interoperability
- Reusability
- Adaptability

Maintenance

This aspect briefs about how well the software has the capabilities to maintain itself in the ever-changing environment:

- Modularity
- Maintainability
- Flexibility
- Scalability

