

Data Structures and Algorithms

Section I – Data Structures

1. What is data structure?

A data structure is a way of organizing data that considers not only the items stored, but also their relationship to each other. Advance knowledge about the relationship between data items allows designing of efficient algorithms for the manipulation of data.

2. List out the areas in which data structures are applied extensively?

- Compiler Design,
- Operating System,
- Database Management System,
- Statistical analysis package,
- Numerical Analysis,
- Graphics,
- Artificial Intelligence,
- Simulation

3. *What are the major data structures used in the following areas : RDBMS, Network data model & Hierarchical data model.*

- RDBMS – Array (i.e. Array of structures)
- Network data model – Graph
- Hierarchical data model – Trees

4. *If you are using C language to implement the heterogeneous linked list, what pointer type will you use?*

The heterogeneous linked list contains different data types in its nodes and we need a link, pointer to connect them. It is not possible to use ordinary pointers for this. So we go for void pointer. Void pointer is capable of storing pointer to any type as it is a generic pointer type.

5. *Minimum number of queues needed to implement a priority queue?*

Two. One queue is used for actual storing of data and another for storing priorities.

6. *What is the data structure used to perform recursion?*

Stack. Because of its LIFO (Last In First Out) property it remembers its 'caller'; so the function knows where to return when the function has to return. Recursion makes use of system stack for storing the return addresses of the function calls.

Every recursive function has its equivalent iterative (non-recursive) function.

Even when such equivalent iterative procedures are written, explicit stack is to be used.

7. *What are the notations used in Evaluation of Arithmetic Expressions using prefix and postfix forms?*

Polish and Reverse Polish notations.

8. *Convert the expression $((A + B) * C - (D - E) ^ (F + G))$ to equivalent Prefix and Postfix notations.*

Prefix Notation:

$^ - * +ABC - DE + FG$

Postfix Notation:
 $AB + C * DE - - FG + ^$

9. *Sorting is not possible by using which of the following methods?*

(a) *Insertion*

(b) *Selection*

(c) *Exchange*

(d) *Deletion*

Answer:

(d)deletion

Explanation:

Using insertion we can perform insertion sort, using selection we can perform selection sort, using exchange we can perform the bubble sort (and other similar sorting methods). But no sorting method can be done just using deletion.

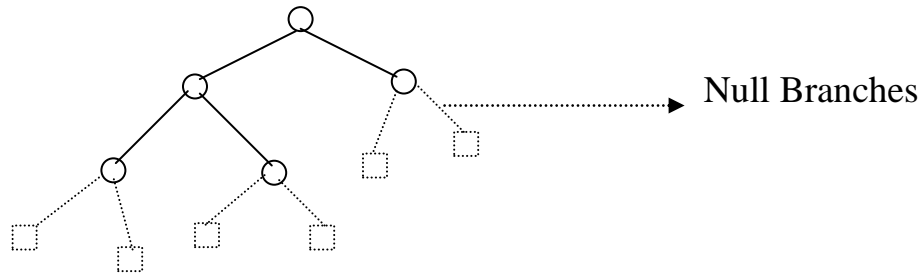
10. *A binary tree with 20 nodes has _____ null branches.*

Answer:

21

Explanation:

Let us take a tree with 5 nodes ($n=5$)



It will have only 6 (ie, $5+1$) null branches. In general, a *binary tree with n nodes has exactly $n+1$ null nodes.*

11. *What are the methods available in storing sequential files ?*

- Straight merging,
- Natural merging,
- Polyphase sort,
- Distribution of Initial runs.

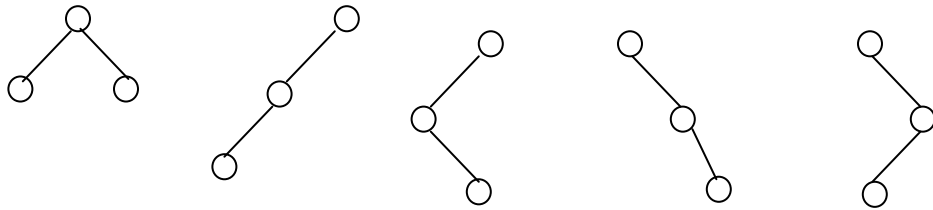
12. How many different trees are possible with 10 nodes ?

Answer:

1014

Explanation:

For example, consider a tree with 3 nodes($n=3$), it will have the maximum combination of 5 different (ie, $2^3 - 3 = 5$) trees.



i

ii

iii

iv

v

In general:

If there are n nodes, there exist $2^n - n$ different trees.

13. List out a few of the applications of tree data-structure in compilers/compiler design.

- The manipulation of Arithmetic expression,
- Symbol Table construction,
- Syntax analysis.

14. *List out a few of the applications that make use of Multilinked Structures.*

- Sparse matrix,
- Index generation.

15. *In tree construction which is the suitable efficient data structure?*

- (a) Array
- (b) Linked list
- (c) Stack
- (d) Queue
- (e) none

Answer:

- (b) Linked list

16. *What type of the algorithm is used in solving the 8 Queens problem?*

Backtracking

17. *In an AVL tree, what is the condition for balancing to be done?*

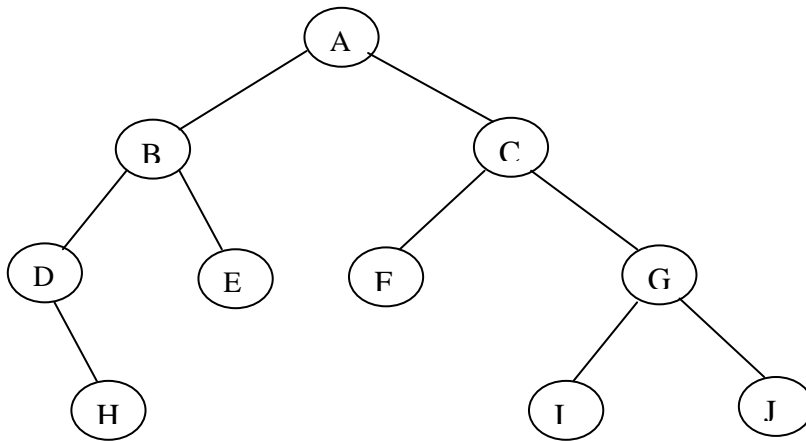
The 'pivotal value'(or the 'Height factor') is greater than 1 or less than-1.

18. *What is the bucket size, when overlapping and collision occur at the same time?*

One. If there is only one entry possible in the bucket, when the collision occurs, there is no way to accommodate the colliding value. This results in the overlapping of values.

19. *Traverse the given tree using Inorder, Preorder and Postorder traversals.*

Given tree:



- Inorder : D H B E A F C I G J
- Preorder: A B D H E C F G I J
- Postorder: H D E B F I J G C A

20. There are 8, 15, 13, 14 nodes were there in 4 different trees. Which of them could have formed a full binary tree?

15.

In general:

There are $2^n - 1$ nodes in a full binary tree.

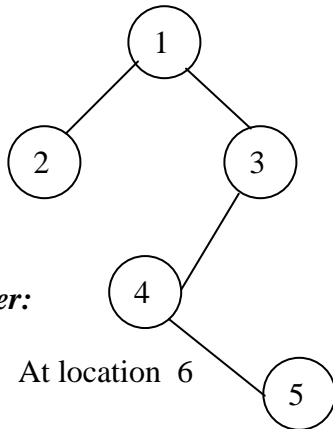
By the method of elimination:

Full binary trees contain *odd* number of nodes. So there cannot be full binary trees with 8 or 14 nodes, so rejected. With 13 nodes you can form a *complete* binary tree but not a full binary tree. So the correct answer is 15.

Note:

Full and Complete binary trees are different. *All full binary trees are complete binary trees but not vice versa.*

21. In the given binary tree, using array, at which location can you store the node 4?



Answer:

At location 6

Explanation:

1	2	3	-	-	4	-	-	5
Root	LC1	RC1	LC2	RC2	LC3	RC3	LC4	RC4

where LCn means Left Child of node n and RCn means Right Child of node n

22. Sort the given values using Quick Sort?

65 70 75 80 85 60 55 50 45

Sorting takes place from the pivot value, which is the first value of the given elements, this is marked bold. The values at the left pointer and right pointer are indicated using ^L and ^R respectively.

65 70^L 75 80 85 60 55 50 45^R

Since pivot is not yet changed the same process is continued after interchanging the values at ^L and ^R positions

65 45 75^L 80 85 60 55 50^R 70

65 45 50 80^L 85 60 55^R 75 70

65 45 50 55 85^L 60^R 80 75 70

65 45 50 55 60^R 85^L 80 75 70

When the L and R pointers cross each other the pivot value is interchanged with the value at right pointer. If the pivot is changed it means that the pivot has occupied its original position in the sorted order (shown in bold italics) and hence two different arrays are formed, one from start of the original array to the pivot position-1 and the other from pivot position+1 to end.

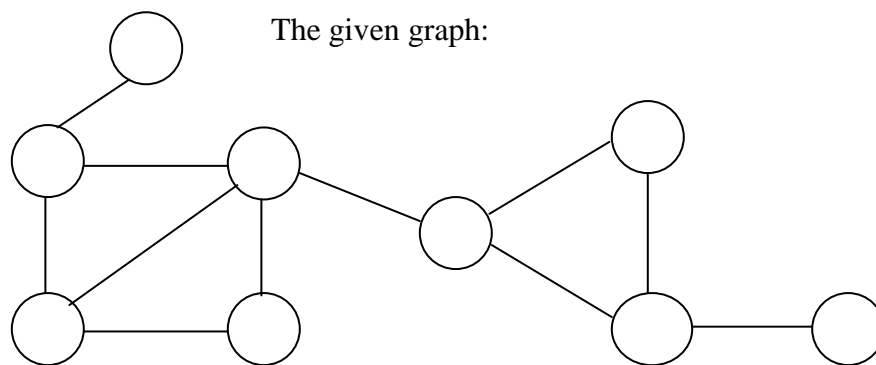
60^L 45 50 55^R **65** **85^L** 80 75 70^R

55^L 45 50^R **60** **65** 70^R 80^L 75 **85**
 50^L 45^R 55 **60** **65** 70 80^L 75^R **85**

In the next pass we get the sorted form of the array.

45 50 55 60 65 70 75 80 85

23. For the given graph, draw the DFS and BFS?



- *BFS*: A X G H P E M Y J
- *DFS*: A X H P E Y M J G

24. Classify the Hashing Functions based on the various methods by which the key value is found.

- Direct method,

- Subtraction method,
- Modulo-Division method,
- Digit-Extraction method,
- Mid-Square method,
- Folding method,
- Pseudo-random method.

25. *What are the types of Collision Resolution Techniques and the methods used in each of the type?*

- Open addressing (closed hashing).

The methods used include:

Overflow block.

- Closed addressing (open hashing)

The methods used include:

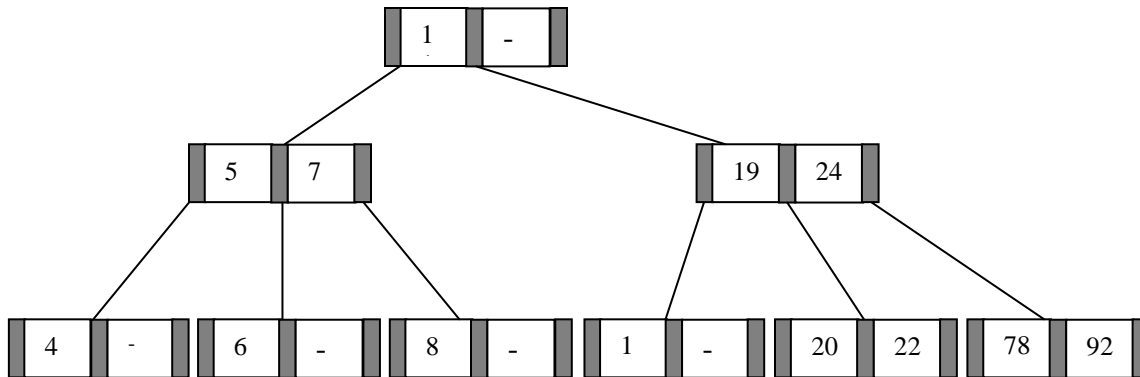
Linked list, Binary tree...

26. *In RDBMS, what is the efficient data structure used in the internal storage representation?*

B+ tree. Because in B+ tree, all the data are stored only in leaf nodes, that makes searching easier. This corresponds to the records that shall be stored in leaf nodes.

27. *Draw the B-tree of order 3 created by inserting the following data arriving in sequence –*

92 24 6 7 11 8 22 4 5 16 19 20 78



28. *Of the following tree structure, which is efficient considering space and time complexities?*

- (a) *Incomplete Binary Tree*
- (b) *Complete Binary Tree*
- (c) *Full Binary Tree*

Answer:

- (b) Complete Binary Tree.

Explanation:

By the method of elimination:

Full binary tree loses its nature when operations of insertions and deletions are done. For incomplete binary trees, extra storage is required and overhead of NULL node

checking takes place. So complete binary tree is the better one since the property of complete binary tree is maintained even after operations like additions and deletions are done on it.

29. *What is a spanning Tree?*

A spanning tree is a tree associated with a network. All the nodes of the graph appear on the tree once. A minimal spanning tree is a spanning tree organized so that the total edge weight between nodes is minimized.

30. *Does the minimal spanning tree of a graph give the shortest distance between any 2 specified nodes?*

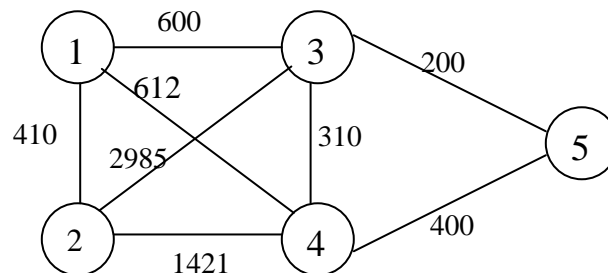
Answer:

No.

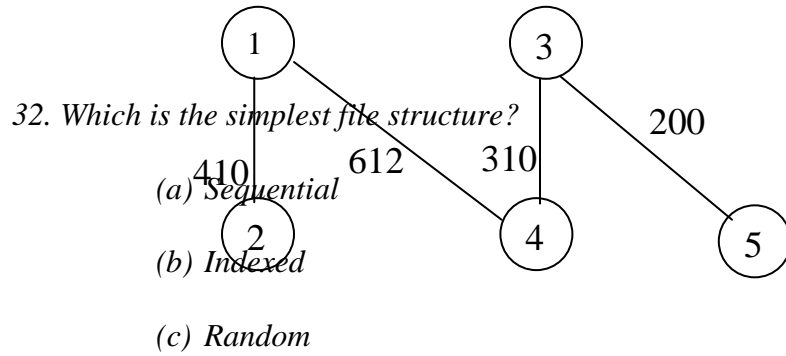
Explanation:

Minimal spanning tree assures that the total weight of the tree is kept at its minimum. But it *doesn't* mean that the distance between any two nodes involved in the minimum-spanning tree is minimum.

31. *Convert the given graph with weighted edges to minimal spanning tree.*



the equivalent minimal spanning tree is:



Answer:

(a) Sequential

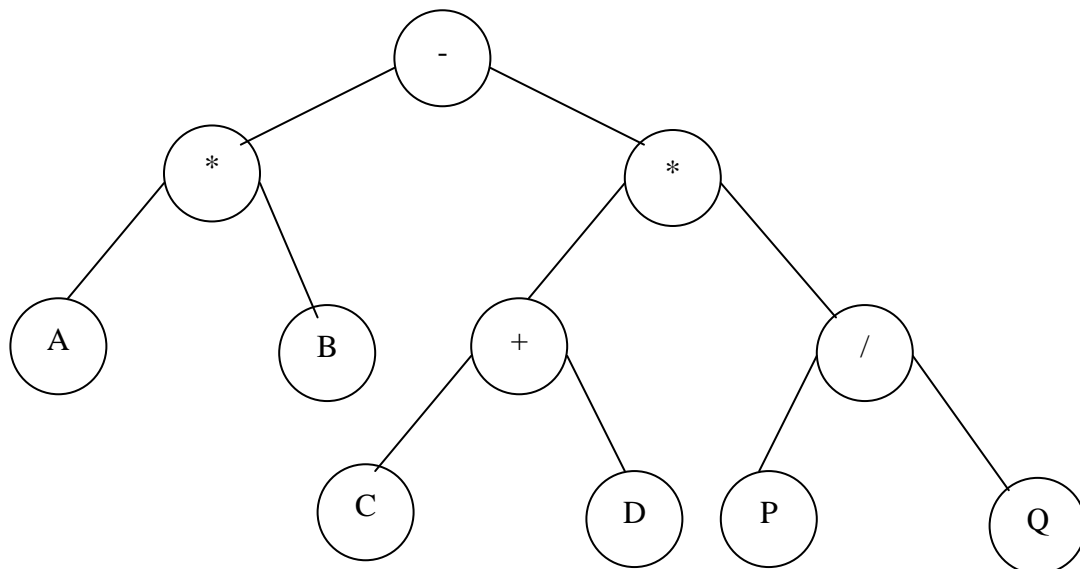
33. Is linked-list a linear or non-linear data structure?

According to Access strategies Linked list is a linear one.

According to Storage, Linked List is a Non-linear one.

34. Draw a binary Tree for the expression :

$$A * B - (C + D) * (P / Q)$$



35. For the following COBOL code, draw the Binary tree?

01 STUDENT_REC.

02 NAME.

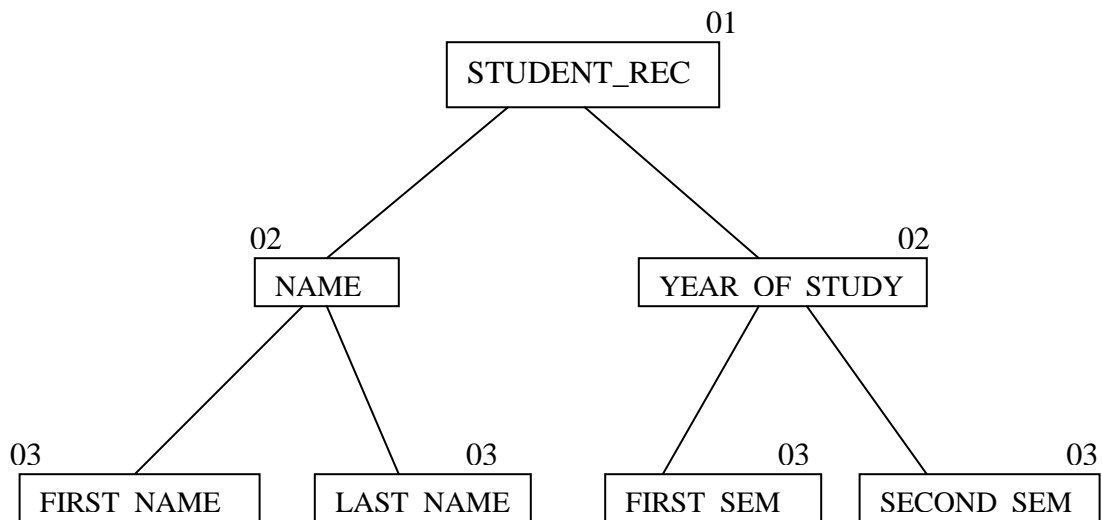
03 FIRST_NAME PIC X(10).

03 LAST_NAME PIC X(10).

02 YEAR_OF_STUDY.

03 FIRST_SEM PIC XX.

03 SECOND_SEM PIC XX.



Section II – Algorithms

1. *What is an ‘algorithm’?*

An algorithm consists of a finite set of steps that may require one or more operations. These operations should be definite and effective. An algorithm should produce one or more output's and may have zero or more inputs.

This consists of five distinct areas:

1. to device algorithms
2. to validate the algorithms
3. to express the algorithms
4. to analyse the algorithms
5. to test the programs for the algorithms

2. *What is a ‘computational procedure’?*

An algorithm that does not terminate is called ‘computational procedure’. Example for such ‘computational procedure’ is an ‘operating system’.

3. *Define - ‘recursive algorithm’.*

An algorithm is said to be recursive if the same algorithm is invoked in the body of the algorithm. Recursive algorithms can be divided into direct recursive and indirect recursive algorithms.

Direct recursive:

An algorithm that calls itself.

Indirect Recursive:

An algorithm 'A' is said to be indirect recursive if it calls another algorithm which in-turn calls algorithm 'A'.

4. *How can you classify performance analysis?*

'Performance analysis' can be classified as:

- i. priori analysis
- ii. posteriori analysis

Priori Analysis:

The bounds of algorithms' computing time are obtained by formulating a function.

Posteriori Analysis:

Testing the actual computation of space and time are recorded while the algorithm is executing.

5. *Define - 'Big- O'.*

For the function $f(n)$

$$f(n) = O(g(n))$$

iff there exist positive constants c and d such that:

$$f(n) \leq c * g(n)$$

for all $n, n \geq d$.

This is defined to be the worst-time complexity of the function $f(n)$.

For example:

$O(n)=3n+2$ because,
 $3n+2 \leq 4n$ for all $n \geq 2$.

6. *Give various computing times and their meaning.*

Few of the important computing times are:

<i>Computing Time</i>	<i>Meaning</i>
$O(1)$: constant computing time
$O(n)$: linear computing time
$O(n*n)$: quadratic computing time
$O(n*n*n)$: cubic computing time
$O(2*2*2*2*.....*n)$: exponential computing time

7. *Give the most important 'basic designs' of algorithms.*

There are five important basic designs for algorithms. They are:

- i. Divide and conquer,
- ii. The greedy method,
- iii. Dynamic programming,
- iv. Back-tracking,
- v. Branch and bound.

8. *How does 'divide and conquer' algorithms work?*

For a function to compute on n inputs the divide and conquer strategy suggests the inputs into a k distinct subsets, $1 < k \leq n$, yielding k sub-problems. These sub-problems must be solved and then a method must be found to combine the sub-solutions into a solution of the whole.

An example for this approach is 'binary search' algorithm. The time complexity of binary search algorithm is $O(\log n)$.

9. *What is Greedy Method?*

The greedy method suggests that one can devise an algorithm that works in stages, considering one input at a time. At each stage, a decision is made regarding whether a particular input is an optimal solution. An example for solution using greedy method is 'knapsack problem'.

10. *What is Dynamic Programming?*

Dynamic Programming is an algorithm design method that can be used when the solution to a problem can be viewed as the result of a sequence of decisions. An example for algorithm using dynamic programming is 'multistage graphs'.

11. *What are the time complexities for the following algorithms?*

Binary search : $O(\log n)$

Finding maximum and minimum for a given set of numbers

: $O(3n/2 - 2)$

Merge Sort : $O(n \log n)$

Insertion Sort	:	$O(n^2)$;
Quick Sort	:	$O(n \log n)$
Selection Sort	:	$O(n^2)$

12. *What is the difference between Merge Sort and Quick sort?*

Both Merge-sort and Quick-sort have same time complexity i.e. $O(n \log n)$. In merge sort the file $a[1:n]$ was divided at its midpoint into sub-arrays which are independently sorted and later merged. Whereas, in quick sort the division into two sub-arrays is made so that the sorted sub-arrays do not need to be merged later.

13. *Is there any optimum solution for Matrix multiplication?*

Yes. Divide and conquer method suggests Strassen's matrix multiplication method to be used. If we follow this method, the time complexity is $O(n^2.81)$ times rather $O(n^3)$ times.

14. *Define 'minimum cost spanning method'.*

Let $G=(V,E)$ be an undirected connected graph. A sub-graph $t=(V, E')$ of G is a spanning tree of G if and only if t is a tree.

To find out minimum cost spanning method we have following method's;

Prim's Algorithm	:	$O(n^2)$
Kruskal's Algorithm	:	$O(e \log e)$

15. *Define 'articulation points'.*

A Vertex V in a connected graph G is an articulation point if and only if the deletion of vertex V together with all edges incident to it disconnects the graph into two or more non-empty Components.

16. *Define 'biconnected graph'.*

A graph G is biconnected if and only if it contains no articulation points.

17. *What are 'explicit' and 'implicit' constraints?*

'Explicit constraints' are rules that restrict each x_i to take on values only from a given set. 'Implicit constraints' are rules that determine which of the tuples in the solution space of i satisfy the criterion function.

18. *Give 'Cook's – theorem'.*

Satisfiability is in P iff $P=NP$

where P is the set of all decision problems solvable by deterministic algorithms in polynomial time and NP is the set of all decision problems solvable by non-deterministic algorithms in a polynomial-time.

RDBMS Concepts

1. *What is database?*

A database is a logically coherent collection of data with some inherent meaning, representing some aspect of real world and which is designed, built and populated with data for a specific purpose.

2. *What is DBMS?*

It is a collection of programs that enables an user to create and maintain a database. In other words it is general-purpose software that provides users with the processes of *defining*, *constructing* and *manipulating* the database for various applications.

3. *What is a Database system?*

The database and DBMS software together is called as Database system.

4. *What are the advantages of DBMS?*

As compared to raw access and storage in files, DBMS has many advantages to use.

It includes:

- Redundancy is controlled.
- Unauthorised access is restricted.
- Providing multiple user interfaces.
- Enforcing integrity constraints.
- Providing backup and recovery.

5. *What are the disadvantage in File Processing System?*

- Data redundancy & inconsistency.
- Difficult in accessing data.
- Data isolation.
- Data integrity.
- Concurrent access is not possible.
- Security Problems.

6. *Describe the three levels of data abstraction?*

There are three levels of abstraction:

- *Physical level:* The lowest level of abstraction describes how data are stored.
- *Logical level:* This is the next higher level of abstraction that describes what data are stored in database and what is the relationship among those data.
- *View level:* The highest level of abstraction describes only part of entire database.

7. *Define the "integrity rules".*

There are two Integrity rules.

- *Entity Integrity*: States that “Primary key cannot have NULL value”
- *Referential Integrity*: States that “Foreign Key can be either a NULL value or should be Primary Key value of other relation”.

8. *What is extension and intension?*

Extension - It is the number of tuples present in a table at any instance. This is time dependent.

Intension - It is a constant value that gives the name, structure of table and the constraints laid on it.

9. *What is System R? What are its two major subsystems?*

System R was designed and developed over a period of 1974-79 at IBM San Jose Research Center. It is a prototype and its purpose was to demonstrate that it is possible to build a Relational System that can be used in a real life environment to solve real life problems, with performance at least comparable to that of existing system.

Its two subsystems are

- Research Storage
- System Relational Data System.

10. *How is the data structure of System R different from the relational structure?*

Unlike Relational systems in System R

- Domains are not supported

- Enforcement of candidate key uniqueness is optional
- Enforcement of entity integrity is optional
- Referential integrity is not enforced

11. *What is Data Independence?*

Data independence means that “the application is independent of the storage structure and access strategy of data”. In other words, The ability to modify the schema definition in one level should not affect the schema definition in the next higher level.

Two types of Data Independence:

- Physical Data Independence: Modification in physical level should not affect the logical level.
- Logical Data Independence: Modification in logical level should affect the view level.

NOTE: Logical Data Independence is more difficult to achieve

12. *What is a view? How it is related to data independence?*

A view may be thought of as a virtual table, that is, a table that does not really exist in its own right but is instead derived from one or more underlying base table. In other words, there is no stored file that direct represents the view instead a definition of view is stored in data dictionary.

Growth and restructuring of base tables is not reflected in views. Thus the view can insulate users from the effects of restructuring and growth in the database. Hence accounts for logical data independence.

13. *What is Data Model?*

A collection of conceptual tools for describing data, data relationships, data semantics and constraints.

14. *What is E-R model?*

This data model is based on real world that consists of basic objects called entities and of relationship among these objects. Entities are described in a database by a set of attributes.

15. *What is Object Oriented model?*

This model is based on collection of objects. An object contains values stored in instance variables with in the object. An object also contains bodies of code that operate on the object. These bodies of code are called methods. Objects that contain same types of values and the same methods are grouped together into classes.

16. *What is an Entity?*

It is a 'thing' in the real world with an independent existence.

17. *What is an Entity type?*

It is a collection (set) of entities that have same attributes.

18. *What is an Entity set?*

It is a collection of all entities of particular entity type in the database.

19. *What is an Extension of entity type?*

The collections of entities of a particular entity type are grouped together into an entity set.

20. *What is Weak Entity set?*

An entity set may not have sufficient attributes to form a primary key, and its primary key comprises of its partial key and primary key of its parent entity, then it is said to be Weak Entity set.

21. *What is an attribute?*

It is a particular property, which describes the entity.

22. *What is a Relation Schema and a Relation?*

A relation Schema denoted by $R(A_1, A_2, \dots, A_n)$ is made up of the relation name R and the list of attributes A_i that it contains. A relation is defined as a set of tuples. Let r be the relation which contains set tuples $(t_1, t_2, t_3, \dots, t_n)$. Each tuple is an ordered list of n -values $t=(v_1, v_2, \dots, v_n)$.

23. *What is degree of a Relation?*

It is the number of attributes of its relation schema.

24. *What is Relationship?*

It is an association among two or more entities.

25. *What is Relationship set?*

The collection (or set) of similar relationships.

26. *What is Relationship type?*

Relationship type defines a set of associations or a relationship set among a given set of entity types.

27. *What is degree of Relationship type?*

It is the number of entity types participating.

25. *What is DDL (Data Definition Language)?*

A data base schema is specified by a set of definitions expressed by a special language called DDL.

26. *What is VDL (View Definition Language)?*

It specifies user views and their mappings to the conceptual schema.

27. *What is SDL (Storage Definition Language)?*

This language is to specify the internal schema. This language may specify the mapping between two schemas.

28. *What is Data Storage - Definition Language?*

The storage structures and access methods used by database system are specified by a set of definition in a special type of DDL called data storage-definition language.

29. *What is DML (Data Manipulation Language)?*

This language that enables the user to access or manipulate data as organised by appropriate data model.

➤ *Procedural DML or Low level:* DML requires a user to specify what data are needed and how to get those data.

➤ *Non-Procedural DML or High level:* DML requires a user to specify what data are needed without specifying how to get those data.

30. *What is DML Compiler?*

It translates DML statements in a query language into low-level instruction that the query evaluation engine can understand.

31. *What is Query evaluation engine?*

It executes low-level instruction generated by compiler.

32. *What is DDL Interpreter?*

It interprets DDL statements and record them in tables containing metadata.

34. *What is Record-at-a-time?*

The Low level or Procedural DML can specify and retrieve each record from a set of records. This retrieve of a record is said to be Record-at-a-time.

35. *What is Set-at-a-time or Set-oriented?*

The High level or Non-procedural DML can specify and retrieve many records in a single DML statement. This retrieve of a record is said to be Set-at-a-time or Set-oriented.

36. *What is Relational Algebra?*

It is procedural query language. It consists of a set of operations that take one or two relations as input and produce a new relation.

37. *What is Relational Calculus?*

It is an applied predicate calculus specifically tailored for relational databases proposed by E.F. Codd. Example for the languages based on it are DSL ALPHA and QUEL.

38. *How does Tuple-oriented relational calculus differ from domain-oriented relational calculus*

The tuple-oriented calculus uses a tuple variables i.e., variable whose only permitted values are tuples of that relation. E.g. QUEL

The domain-oriented calculus has domain variables i.e., variables that range over the underlying domains instead of over relation. E.g. ILL, DEDUCE.

39. *What is normalization?*

It is a process of analysing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties

- Minimizing redundancy
- Minimizing insertion, deletion and updation anomalies.

40. *What is Functional Dependency?*

A Functional dependency is denoted by $X \twoheadrightarrow Y$ between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuple that can form a relation state r of R. The constraint is for any two tuples t1 and t2 in r if $t1[X] = t2[X]$ then they have $t1[Y] = t2[Y]$. This means the value of X component of a tuple uniquely determines the value of component Y.

41. *When is a functional dependency F said to be minimal?*

- Every dependency in F has a single attribute for its right hand side.
- We cannot replace any dependency $X \twoheadrightarrow A$ in F with a dependency $Y \twoheadrightarrow A$ where Y is a proper subset of X and still have a set of dependency that is equivalent to F.
- We cannot remove any dependency from F and still have a set of dependency that is equivalent to F.

42. *What is Multivalued dependency?*

Multivalued dependency denoted by $X \twoheadrightarrow Y$ specified on relation schema R, where X and Y are both subsets of R, specifies the following constraint on any relation r of R: if two tuples t1 and t2 exist in r such that $t1[X] = t2[X]$ then t3 and t4 should also exist in r with the following properties

- $t3[X] = t4[X] = t1[X] = t2[X]$
- $t3[Y] = t1[Y]$ and $t4[Y] = t2[Y]$
- $t3[Z] = t2[Z]$ and $t4[Z] = t1[Z]$

where $[Z = (R - (X \cup Y))]$

43. *What is Lossless join property?*

It guarantees that the spurious tuple generation does not occur with respect to relation schemas after decomposition.

44. *What is 1NF (Normal Form)?*

The domain of attribute must include only atomic (simple, indivisible) values.

45. *What is meant by Fully Functional Dependency?*

A functional dependency $X \twoheadrightarrow Y$ is full functional dependency if removal of any attribute A from X means that the dependency does not hold any more.

46. *What is 2NF?*

A relation schema R is in 2NF if it is in 1NF and every non-prime attribute A in R is fully functionally dependent on primary key.

47. *What is 3NF?*

A relation schema R is in 3NF if it is in 2NF and for every FD $X \rightarrow A$ either of the following is true

- X is a Super-key of R.
- A is a prime attribute of R.

In other words, if every non prime attribute is non-transitively dependent on primary key.

48. *What is BCNF (Boyce-Codd Normal Form)?*

A relation schema R is in BCNF if it is in 3NF and satisfies an additional constraint that for every FD $X \rightarrow A$, X must be a candidate key.

49. *What is 4NF?*

A relation schema R is said to be in 4NF if for every Multivalued dependency $X \twoheadrightarrow Y$ that holds over R, one of following is true

- X is subset or equal to (or) $XY = R$.
- X is a super key.

50. *What is 5NF?*

A Relation schema R is said to be 5NF if for every join dependency $\{R_1, R_2, \dots, R_n\}$ that holds R, one of the following is true :

- $R_i = R$ for some i.
- The join dependency is implied by the set of FD, over R in which the left side is key of R.

51. *What is Domain-Key Normal Form?*

A relation is said to be in DKNF if all constraints and dependencies that should hold on the the constraint can be enforced by simply enforcing the domain constraint and key constraint on the relation.

52. *What are partial, alternate,, artificial, compound and natural key?*

Partial Key:

It is a set of attributes that can uniquely identify weak entities and that are related to same owner entity. It is sometime called as Discriminator.

Alternate Key:

All Candidate Keys excluding the Primary Key are known as Alternate Keys.

Artificial Key:

If no obvious key, either stand alone or compound is available, then the last resort is to simply create a key, by assigning a unique number to each record or occurrence.

Then this is known as developing an artificial key.

Compound Key:

If no single data element uniquely identifies occurrences within a construct, then combining multiple elements to create a unique identifier for the construct is known as creating a compound key.

Natural Key:

When one of the data elements stored within a construct is utilized as the primary key, then it is called the natural key.

53. *What is indexing and what are the different kinds of indexing?*

Indexing is a technique for determining how quickly specific data can be found.

Types:

- Binary search style indexing
- B-Tree indexing
- Inverted list indexing
- Memory resident table
- Table indexing

54. *What is system catalog or catalog relation? How is better known as?*

A RDBMS maintains a description of all the data that it contains, information about every relation and index that it contains. This information is stored in a collection of relations maintained by the system called metadata. It is also called data dictionary.

55. *What is meant by query optimization?*

The phase that identifies an efficient execution plan for evaluating a query that has the least estimated cost is referred to as query optimization.

56. *What is join dependency and inclusion dependency?*

Join Dependency:

A Join dependency is generalization of Multivalued dependency. A JD $\{R_1, R_2, \dots, R_n\}$ is said to hold over a relation R if $R_1, R_2, R_3, \dots, R_n$ is a lossless-join decomposition of R. There is no set of sound and complete inference rules for JD.

Inclusion Dependency:

An Inclusion Dependency is a statement of the form that some columns of a relation are contained in other columns. A foreign key constraint is an example of inclusion dependency.

57. *What is durability in DBMS?*

Once the DBMS informs the user that a transaction has successfully completed, its effects should persist even if the system crashes before all its changes are reflected on disk. This property is called durability.

58. *What do you mean by atomicity and aggregation?*

Atomicity:

Either all actions are carried out or none are. Users should not have to worry about the effect of incomplete transactions. DBMS ensures this by undoing the actions of incomplete transactions.

Aggregation:

A concept which is used to model a relationship between a collection of entities and relationships. It is used when we need to express a relationship among relationships.

59. *What is a Phantom Deadlock?*

In distributed deadlock detection, the delay in propagating local information might cause the deadlock detection algorithms to identify deadlocks that do not really exist. Such situations are called phantom deadlocks and they lead to unnecessary aborts.

60. *What is a checkpoint and When does it occur?*

A Checkpoint is like a snapshot of the DBMS state. By taking checkpoints, the DBMS can reduce the amount of work to be done during restart in the event of subsequent crashes.

61. *What are the different phases of transaction?*

Different phases are

- Analysis phase
- Redo Phase
- Undo phase

62. *What do you mean by flat file database?*

It is a database in which there are no programs or user access languages. It has no cross-file capabilities but is user-friendly and provides user-interface management.

63. *What is "transparent DBMS"?*

It is one, which keeps its Physical Structure hidden from user.

64. *Brief theory of Network, Hierarchical schemas and their properties*

Network schema uses a graph data structure to organize records example for such a database management system is CTCG while a hierarchical schema uses a tree data structure example for such a system is IMS.

65. *What is a query?*

A query with respect to DBMS relates to user commands that are used to interact with a data base. The query language can be classified into data definition language and data manipulation language.

66. *What do you mean by Correlated subquery?*

Subqueries, or nested queries, are used to bring back a set of rows to be used by the parent query. Depending on how the subquery is written, it can be executed once for the parent query or it can be executed once for each row returned by the parent query. If the subquery is executed for each row of the parent, this is called a *correlated subquery*.

A correlated subquery can be easily identified if it contains any references to the parent subquery columns in its WHERE clause. Columns from the subquery cannot be referenced anywhere else in the parent query. The following example demonstrates a non-correlated subquery.

E.g. Select * From CUST Where '10/03/1990' IN (Select ODATE From ORDER
Where CUST.CNUM = ORDER.CNUM)

67. *What are the primitive operations common to all record management systems?*

Addition, deletion and modification.

68. *Name the buffer in which all the commands that are typed in are stored*

'Edit' Buffer

69. *What are the unary operations in Relational Algebra?*

PROJECTION and SELECTION.

70. *Are the resulting relations of PRODUCT and JOIN operation the same?*

No.

PRODUCT: Concatenation of every row in one relation with every row in another.

JOIN: Concatenation of rows from one relation and related rows from another.

71. *What is RDBMS KERNEL?*

Two important pieces of RDBMS architecture are the kernel, which is the software, and the data dictionary, which consists of the system-level data structures used by the kernel to manage the database

You might think of an RDBMS as an operating system (or set of subsystems), designed specifically for controlling data access; its primary functions are storing, retrieving, and securing data. An RDBMS maintains its own list of authorized users and their associated privileges; manages memory caches and paging; controls locking for concurrent resource usage; dispatches and schedules user requests; and manages space usage within its table-space structures

72. *Name the sub-systems of a RDBMS*

I/O, Security, Language Processing, Process Control, Storage Management, Logging and Recovery, Distribution Control, Transaction Control, Memory Management, Lock Management

73. *Which part of the RDBMS takes care of the data dictionary? How*

Data dictionary is a set of tables and database objects that is stored in a special area of the database and maintained exclusively by the kernel.

74. *What is the job of the information stored in data-dictionary?*

The information in the data dictionary validates the existence of the objects, provides access to them, and maps the actual physical storage location.

75. *Not only RDBMS takes care of locating data it also*

determines an optimal access path to store or retrieve the data

76. *How do you communicate with an RDBMS?*

You communicate with an RDBMS using Structured Query Language (SQL)

77. *Define SQL and state the differences between SQL and other conventional programming Languages*

SQL is a nonprocedural language that is designed specifically for data access operations on normalized relational database structures. The primary difference between SQL and other conventional programming languages is that SQL statements specify what data operations should be performed rather than how to perform them.

78. *Name the three major set of files on disk that compose a database in Oracle*

There are three major sets of files on disk that compose a database. All the files are binary. These are

- Database files
- Control files
- Redo logs

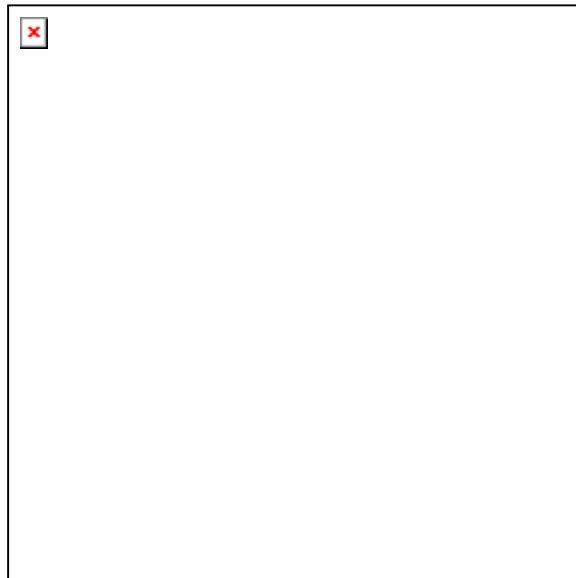
The most important of these are the database files where the actual data resides. The control files and the redo logs support the functioning of the architecture itself.

All three sets of files must be present, open, and available to Oracle for any data on the database to be useable. Without these files, you cannot access the database, and the database administrator might have to recover some or all of the database using a backup, if there is one.

79. *What is an Oracle Instance?*

The Oracle system processes, also known as Oracle background processes, provide functions for the user processes—functions that would otherwise be done by the user processes themselves

Oracle database-wide system memory is known as the SGA, the *system global area* or *shared global area*. The data and control structures in the SGA are shareable, and all the Oracle background processes and user processes can use them.



The combination of the SGA and the

Oracle background processes is known as an *Oracle instance*

80. *What are the four Oracle system processes that must always be up and running for the database to be useable*

The four Oracle system processes that must always be up and running for the database to be useable include *DBWR* (Database Writer), *LGWR* (Log Writer), *SMON* (System Monitor), and *PMON* (Process Monitor).

81. *What are database files, control files and log files. How many of these files should a database have at least? Why?*

Database Files

The database files hold the actual data and are typically the largest in size. Depending on their sizes, the tables (and other objects) for all the user accounts can go in one database file—but that's not an ideal situation because it does not make the database structure very flexible for controlling access to storage for different users, putting the database on different disk drives, or backing up and restoring just part of the database.

You must have at least one database file but usually, more than one files are used. In terms of accessing and using the data in the tables and other objects, the number (or location) of the files is immaterial.

The database files are fixed in size and never grow bigger than the size at which they were created

Control Files

The control files and redo logs support the rest of the architecture. Any database must have at least one control file, although you typically have more than one to guard against loss. The control file records the name of the database, the date and time it was created, the location of the database and redo logs, and the synchronization information to ensure that all three sets of files are always in step. Every time you add a new database or redo log file to the database, the information is recorded in the control files.

Redo Logs

Any database must have at least two redo logs. These are the journals for the database; the redo logs record all changes to the user objects or system objects. If any

type of failure occurs, the changes recorded in the redo logs can be used to bring the database to a consistent state without losing any committed transactions. In the case of non-data loss failure, Oracle can apply the information in the redo logs automatically without intervention from the DBA.

The redo log files are fixed in size and never grow dynamically from the size at which they were created.

82. *What is ROWID?*

The ROWID is a unique database-wide physical address for every row on every table. Once assigned (when the row is first inserted into the database), it never changes until the row is deleted or the table is dropped.

The ROWID consists of the following three components, the combination of which uniquely identifies the physical storage location of the row.

- Oracle database file number, which contains the block with the rows
- Oracle block address, which contains the row
- The row within the block (because each block can hold many rows)

The ROWID is used internally in indexes as a quick means of retrieving rows with a particular key value. Application developers also use it in SQL statements as a quick way to access a row once they know the ROWID

83. *What is Oracle Block? Can two Oracle Blocks have the same address?*

Oracle "formats" the database files into a number of Oracle blocks when they are first created—making it easier for the RDBMS software to manage the files and easier to read data into the memory areas.

The block size should be a multiple of the operating system block size. Regardless of the block size, the entire block is not available for holding data; Oracle takes up some space to manage the contents of the block. This block header has a minimum size, but it can grow.

These Oracle blocks are the smallest unit of storage. Increasing the Oracle block size can improve performance, but it should be done only when the database is first created.

Each Oracle block is numbered sequentially for each database file starting at 1. Two blocks can have the same block address if they are in different database files.

84. *What is database Trigger?*

A database trigger is a PL/SQL block that can be defined to automatically execute for insert, update, and delete statements against a table. The trigger can be defined to execute once for the entire statement or once for every row that is inserted, updated, or deleted. For any one table, there are twelve events for which you can define database triggers. A database trigger can call database procedures that are also written in PL/SQL.

85. *Name two utilities that Oracle provides, which are used for backup and recovery.*

Along with the RDBMS software, Oracle provides two utilities that you can use to back up and restore the database. These utilities are *Export* and *Import*.

The *Export utility* dumps the definitions and data for the specified part of the database to an operating system binary file. The *Import utility* reads the file produced by an export, recreates the definitions of objects, and inserts the data.

If Export and Import are used as a means of backing up and recovering the database, all the changes made to the database cannot be recovered since the export was performed. The best you can do is recover the database to the time when the export was last performed.

86. *What are stored-procedures? And what are the advantages of using them.*

Stored procedures are database objects that perform a user defined operation. A stored procedure can have a set of compound SQL statements. A stored procedure executes the SQL commands and returns the result to the client. Stored procedures are used to reduce network traffic.

87. *How are exceptions handled in PL/SQL?*

PL/SQL exception handling is a mechanism for dealing with run-time errors encountered during procedure execution. Use of this mechanism enables execution to continue if the error is not severe enough to cause procedure termination.

The exception handler must be defined within a subprogram specification. Errors cause the program to raise an exception with a transfer of control to the exception-handler block. After the exception handler executes, control returns to the block in which the handler was defined. If there are no more executable statements in the block, control returns to the caller.

User-Defined Exceptions

PL/SQL enables the user to define exception handlers in the declarations area of subprogram specifications. User accomplishes this by naming an exception as in the following example:

```
ot_failure EXCEPTION;
```

In this case, the exception name is `ot_failure`. Code associated with this handler is written in the `EXCEPTION` specification area as follows:

```
EXCEPTION

    when OT_FAILURE then

        out_status_code := g_out_status_code;

        out_msg         := g_out_msg;
```

The following is an example of a subprogram exception:

```
EXCEPTION

    when NO_DATA_FOUND then

        g_out_status_code := 'FAIL';

        RAISE ot_failure;
```

Within this exception is the `RAISE` statement that transfers control back to the `ot_failure` exception handler. This technique of raising the exception is used to invoke all user-defined exceptions.

System-Defined Exceptions

Exceptions internal to PL/SQL are raised automatically upon error. For example `NO_DATA_FOUND` is a system-defined exception. There is also a catch-all exception

named *OTHERS* that traps all errors for which specific error handling has not been established.

88. *Does PL/SQL support "overloading"? Explain*

The concept of *overloading* in PL/SQL relates to the idea that you can define procedures and functions with the same name. PL/SQL does not look only at the referenced name, however, to resolve a procedure or function call. The count and data types of formal parameters are also considered.

PL/SQL also attempts to resolve any procedure or function calls in locally defined packages before looking at globally defined packages or internal functions. To further ensure calling the proper procedure, you can use the dot notation. Prefacing a procedure or function name with the package name fully qualifies any procedure or function reference.

89. *Tables derived from the ERD*

- a) Are totally unnormalised
- b) Are always in 1NF
- c) Can be further denormalised
- d) May have multi-valued attributes

Answer:

- (b) Are always in 1NF

90. *Spurious tuples may occur due to*

(i) *Bad normalization*

(ii) *Theta joins*

(iii) *Updating tables from join*

i & ii b) ii & iii c) i & iii d) ii & iii

Answer:

i & iii

Explanation:

Theta joins are joins made on keys that are not primary keys.

91. *A B C is a set of attributes. The functional dependency is as follows*

$AB \rightarrow B$ $AC \rightarrow C$ $C \rightarrow B$

a) is in 1NF

b) is in 2NF

c) is in 3NF

d) is in BCNF

Answer:

(a) is in 1NF since $(AC)^+ = \{ A, B, C \}$ hence AC is the primary key. Since $C \rightarrow B$ is a FD given, where neither C is a Key nor B is a prime attribute, this it is not in 3NF. Further B is not functionally dependent on key AC thus it is not in 2NF. Thus the given FDs is in 1NF.

92. *In mapping of ERD to DFD*

entities in ERD should correspond to an existing entity/store in DFD

- b) entity in DFD is converted to attributes of an entity in ERD
- c) relations in ERD has 1 to 1 correspondence to processes in DFD
- d) relationships in ERD has 1 to 1 correspondence to flows in DFD

Answer:

(entities in ERD should correspond to an existing entity/store in DFD

93. *A dominant entity is the entity*

- a) on the N side in a 1 : N relationship
- b) on the 1 side in a 1 : N relationship
- c) on either side in a 1 : 1 relationship
- d) nothing to do with 1 : 1 or 1 : N relationship

Answer:

(b) on the 1 side in a 1 : N relationship

94. *Select 'NORTH', CUSTOMER From CUST_DTLS Where REGION = 'N' Order By
CUSTOMER Union Select 'EAST', CUSTOMER From CUST_DTLS Where REGION = 'E'
Order By CUSTOMER*

The above is

- a) Not an error
- b) Error - the string in single quotes 'NORTH' and 'SOUTH'
- c) Error - the string should be in double quotes
- d) Error - ORDER BY clause

Answer:

(d) Error - the ORDER BY clause. Since ORDER BY clause cannot be used in UNIONS

95. *What is Storage Manager?*

It is a program module that provides the interface between the low-level data stored in database, application programs and queries submitted to the system.

96. *What is Buffer Manager?*

It is a program module, which is responsible for fetching data from disk storage into main memory and deciding what data to be cache in memory.

97. *What is Transaction Manager?*

It is a program module, which ensures that database, remains in a consistent state despite system failures and concurrent transaction execution proceeds without conflicting.

98. *What is File Manager?*

It is a program module, which manages the allocation of space on disk storage and data structure used to represent information stored on a disk.

99. *What is Authorization and Integrity manager?*

It is the program module, which tests for the satisfaction of integrity constraint and checks the authority of user to access data.

100. *What are stand-alone procedures?*

Procedures that are not part of a package are known as stand-alone because they independently defined. A good example of a stand-alone procedure is one written in a SQL*Forms application. These types of procedures are not available for reference from other Oracle tools. Another limitation of stand-alone procedures is that they are compiled at run time, which slows execution.

101. *What are cursors give different types of cursors.*

PL/SQL uses cursors for all database information accesses statements. The language supports the use two types of cursors

➤ *Implicit*

➤ *Explicit*

102. *What is cold backup and hot backup (in case of Oracle)?*

➤ *Cold Backup:*

It is copying the three sets of files (database files, redo logs, and control file) when the instance is shut down. This is a straight file copy, usually from the disk directly to tape. You must shut down the instance to guarantee a consistent copy.

If a cold backup is performed, the only option available in the event of data file loss is restoring all the files from the latest backup. All work performed on the database since the last backup is lost.

➤ *Hot Backup:*

Some sites (such as worldwide airline reservations systems) cannot shut down the database while making a backup copy of the files. The cold backup is not an available option.

So different means of backing up database must be used — the hot backup. Issue a SQL command to indicate to Oracle, on a tablespace-by-tablespace basis, that the files of the tablespace are to be backed up. The users can continue to make full use of the files, including making changes to the data. Once the user has indicated that he/she wants to back up the tablespace files, he/she can use the operating system to copy those files to the desired backup destination.

The database must be running in ARCHIVELOG mode for the hot backup option.

If a data loss failure does occur, the lost database files can be restored using the hot backup and the online and offline redo logs created since the backup was done. The database is restored to the most consistent state without any loss of committed transactions.

103. *What are Armstrong rules? How do we say that they are complete and/or sound*

The well-known inference rules for FDs

➤ Reflexive rule :

If Y is subset or equal to X then $X \twoheadrightarrow Y$.

➤ Augmentation rule:

If $X \twoheadrightarrow Y$ then $XZ \twoheadrightarrow YZ$.

➤ Transitive rule:

If $\{X \twoheadrightarrow Y, Y \twoheadrightarrow Z\}$ then $X \twoheadrightarrow Z$.

➤ Decomposition rule :

If $X \twoheadrightarrow YZ$ then $X \twoheadrightarrow Y$.

➤ Union or Additive rule:

If $\{X \twoheadrightarrow Y, X \twoheadrightarrow Z\}$ then $X \twoheadrightarrow YZ$.

➤ Pseudo Transitive rule :

If $\{X \twoheadrightarrow Y, WY \twoheadrightarrow Z\}$ then $WX \twoheadrightarrow Z$.

Of these the first three are known as Armstrong Rules. They are sound because it is enough if a set of FDs satisfy these three. They are called complete because using these three rules we can generate the rest all inference rules.

104. *How can you find the minimal key of relational schema?*

Minimal key is one which can identify each tuple of the given relation schema uniquely. For finding the minimal key it is required to find the closure that is the set of all attributes that are dependent on any given set of attributes under the given set of functional dependency.

Algo. I Determining X^+ , closure for X, given set of FDs F

1. Set $X^+ = X$
2. Set Old $X^+ = X^+$
3. For each FD $Y \twoheadrightarrow Z$ in F and if Y belongs to X^+ then add Z to X^+
4. Repeat steps 2 and 3 until Old $X^+ = X^+$

Algo.II Determining minimal K for relation schema R, given set of FDs F

1. Set K to R that is make K a set of all attributes in R

2. For each attribute A in K
 - a. Compute $(K - A)^+$ with respect to F
 - b. If $(K - A)^+ = R$ then set $K = (K - A)^+$

105. *What do you understand by dependency preservation?*

Given a relation R and a set of FDs F, dependency preservation states that the closure of the union of the projection of F on each decomposed relation Ri is equal to the closure of F. i.e.,

$$((\Pi_{R_1}(F)) \cup \dots \cup (\Pi_{R_n}(F)))^+ = F^+$$

if decomposition is not dependency preserving, then some dependency is lost in the decomposition.

106. *What is meant by Proactive, Retroactive and Simultaneous Update.*

Proactive Update:

The updates that are applied to database before it becomes effective in real world .

Retroactive Update:

The updates that are applied to database after it becomes effective in real world .

Simultaneous Update:

The updates that are applied to database at the same time when it becomes effective in real world .

OOps and OOAD Concepts

1. *Name some pure object oriented languages.*

- Smalltalk,
- Java,
- Eiffel,
- Sather.

2. *What do you mean by the words 'Abstraction', 'Separation', 'Composition', and 'Generalization'?*

Abstraction:

Simplifying the description of a real world entity to its essentials.

Separation:

Treating what an entity does and how it does it independently of each other.

Composition:

Building complex whole components by assembling simpler parts in one of the two ways, *Association and aggregation*.

Generalization:

Identifying common elements in an entity.

3. *What is information hiding?*

Information hiding is a mechanism that separates the implementation of the class from its user.

4. *Differentiate between the message and method.*

Message

Objects communicate by sending messages to each other.

A message is sent to invoke a method.

Method

Provides response to a message.

It is an implementation of an operation.

5. *What is the interface of a class?*

The interface of the class is the view provided to the outside world, which hides its internal structure and behaviour.

6. *What is an adaptor class or Wrapper class?*

A class that has no functionality of its own. Its member functions hide the use of a third party software component or an object with the non-compatible interface or a non-object-oriented implementation.

7. *What is a node class?*

A node class is a class that,

- relies on the base class for services and implementation,
- provides a wider interface to the users than its base class,
- relies primarily on virtual functions in its public interface
- depends on all its direct and indirect base class
- can be understood only in the context of the base class
- can be used as base for further derivation
- can be used to create objects.

A node class is a class that has added new services or functionality beyond the services inherited from its base class.

8. *What is an orthogonal base class?*

If two base classes have no overlapping methods or data they are said to be independent of, or orthogonal to each other. Orthogonal in the sense means that two classes operate in different dimensions and do not interfere with each other in any way. The same derived class may inherit such classes with no difficulty.

9. *What is a container class? What are the types of container classes?*

A container class is a class that is used to hold objects in memory or external storage. A container class acts as a generic holder. A container class has a predefined behavior and a well-known interface. A container class is a supporting class whose purpose is to hide the topology used for maintaining the list of objects in memory. When a container class contains a group of mixed objects, the container is called a

heterogeneous container; when the container is holding a group of objects that are all the same, the container is called a homogeneous container.

10. *What is a protocol class?*

An abstract class is a protocol class if:

- it neither contains nor inherits from classes that contain member data, non-virtual functions, or private (or protected) members of any kind.
- it has a non-inline virtual destructor defined with an empty implementation,
- all member functions other than the destructor including inherited functions, are declared pure virtual functions and left undefined.

11. *What is a mixin class?*

A class that provides some but not all of the implementation for a virtual base class is often called mixin. Derivation done just for the purpose of redefining the virtual functions in the base classes is often called mixin inheritance. Mixin classes typically don't share common bases.

12. *What is a concrete class?*

A concrete class is used to define a useful object that can be instantiated as an automatic variable on the program stack. The implementation of a concrete class is defined. The concrete class is not intended to be a base class and no attempt to minimize dependency on other classes in the implementation or behavior of the class.

13. *What is the handle class?*

A handle is a class that maintains a pointer to an object that is programmatically accessible through the public interface of the handle class.

In case of abstract classes, unless one manipulates the objects of these classes through pointers and references, the benefits of the virtual functions are lost. User code may become dependent on details of implementation classes because an abstract type cannot be allocated statically or on the stack without its size being known. Using pointers or references implies that the burden of memory management falls on the user. Another limitation of abstract class object is of fixed size. Classes however are used to represent concepts that require varying amounts of storage to implement them.

A popular technique for dealing with these issues is to separate what is used as a single object in two parts: a handle providing the user interface and a representation holding all or most of the object's state. The connection between the handle and the representation is typically a pointer in the handle. Often, handles have a bit more data than the simple representation pointer, but not much more. Hence the layout of the handle is typically stable, even when the representation changes and also that handles are small enough to move around relatively freely so that the user needn't use the pointers and the references.

14. *What is an action class?*

The simplest and most obvious way to specify an action in C++ is to write a function. However, if the action has to be delayed, has to be transmitted 'elsewhere' before being performed, requires its own data, has to be combined with other actions, etc

then it often becomes attractive to provide the action in the form of a class that can execute the desired action and provide other services as well. Manipulators used with iostreams is an obvious example.

A common form of action class is a simple class containing just one virtual function.

```
class Action{
    public:
        virtual int do_it( int )=0;
        virtual ~Action( );
}
```

Given this, we can write code say a member that can store actions for later execution without using pointers to functions, without knowing anything about the objects involved, and without even knowing the name of the operation it invokes. For example:

```
class write_file : public Action{
    File& f;
    public:
        int do_it(int){
            return f.write( ).succeed( );
        }
};

class error_message: public Action{
    response_box db(message.cstr( ),"Continue","Cancel","Retry");
```

```
switch (db.getresponse()) {  
    case 0: return 0;  
    case 1: abort();  
    case 2: current_operation.redo();return 1;  
}  
};
```

A user of the Action class will be completely isolated from any knowledge of derived classes such as write_file and error_message.

15. *What are seed classes?*

In C++, you design classes to fulfill certain goals. Usually you start with a sketchy idea of class requirements, filling in more and more details as the project matures. Often you wind up with two classes that have certain similarities. To avoid duplicating code in these classes, you should split up the classes at this point, relegating the common features to a parent and making separate derived classes for the different parts. Classes that are made only for the purpose of sharing code in derived classes are called seed classes.

16. *What is an accessor?*

An accessor is a class operation that does not modify the state of an object. The accessor functions need to be declared as *const* operations

17. *What is an inspector?*

Messages that return the value of an attribute are called inspectors.

18. *What is a modifier?*

A modifier, also called a modifying function is a member function that changes the value of at least one data member. In other words, an operation that modifies the state of an object. Modifiers are also known as 'mutators'.

19. *What is a predicate?*

A predicate is a function that returns a *bool* value.

20. *What is a facilitator?*

A facilitator causes an object to perform some action or service.

21. *State the "Rule of minimality" and its corollary?*

The rule of minimality states that unless a behavior is needed, it shouldn't be part of the ADT.

Corollary of the rule of minimality: If the function or operator can be defined such that, it is not a member. This practice makes a non-member function or operator generally independent of changes to the class's implementation.

22. *What is reflexive association?*

The 'is-a' is called a reflexive association because the reflexive association permits classes to bear the is-a association not only with their super-classes but also with

themselves. It differs from a 'specializes-from' as 'specializes-from' is usually used to describe the association between a super-class and a sub-class. For example:

Printer is-a printer.

23. *What is slicing?*

Slicing means that the data added by a subclass are discarded when an object of the subclass is passed or returned by value or from a function expecting a base class object.

Consider the following class declaration:

```
class base{
    ...
    base& operator =(const base&);
    base (const base&);
}
void fun( ){
    base e=m;
    e=m;
}
```

As base copy functions don't know anything about the derived only the base part of the derived is copied. This is commonly referred to as slicing. One reason to pass objects of classes in a hierarchy is to avoid slicing. Other reasons are to preserve polymorphic behavior and to gain efficiency.

24. *What is a Null object?*

It is an object of some class whose purpose is to indicate that a real object of that class does not exist. One common use for a null object is a return value from a member function that is supposed to return an object with some specified properties but cannot find such an object.

25. *Define precondition and post-condition to a member function.*

Precondition:

A precondition is a condition that must be true on entry to a member function. A class is used correctly if preconditions are never false. An operation is not responsible for doing anything sensible if its precondition fails to hold.

For example, the interface invariants of *stack class* say nothing about pushing yet another element on a stack that is already full. We say that *isful()* is a precondition of the *push* operation.

Post-condition:

A post-condition is a condition that must be true on exit from a member function if the precondition was valid on entry to that function. A class is implemented correctly if post-conditions are never false.

For example, after pushing an element on the stack, we know that *isempty()* must necessarily hold. This is a post-condition of the *push* operation.

26. *What is class invariant?*

A class invariant is a condition that defines all valid states for an object. It is a logical condition to ensure the correct working of a class. Class invariants must hold when an object is created, and they must be preserved under all operations of the class. In particular all class invariants are both preconditions and post-conditions for all operations or member functions of the class.

27. *What are the conditions that have to be met for a condition to be an invariant of the class?*

- The condition should hold at the end of every constructor.
- The condition should hold at the end of every mutator(non-const) operation.

28. *What are proxy objects?*

Objects that points to other objects are called proxy objects or surrogates. Its an object that provides the same interface as its server object but does not have any functionality. During a method invocation, it routes data to the true server object and sends back the return value to the object.

```
template<class T> class Array2D{  
    public:  
        class Array1D{  
            public:  
                T& operator[] (int index);  
                const T& operator[] (int index) const;
```

```

        ...
    };
    Array1D operator[] (int index);
    const Array1D operator[] (int index) const;
    ...
};

```

The following then becomes legal:

```

Array2D<float>data(10,20);
.....
cout<<data[3][6]; // fine

```

Here `data[3]` yields an `Array1D` object and the operator `[]` invocation on that object yields the float in position(3,6) of the original two dimensional array. Clients of the `Array2D` class need not be aware of the presence of the `Array1D` class. Objects of this latter class stand for one-dimensional array objects that, conceptually, do not exist for clients of `Array2D`. Such clients program as if they were using real, live, two-dimensional arrays. Each `Array1D` object stands for a one-dimensional array that is absent from a conceptual model used by the clients of `Array2D`. In the above example, `Array1D` is a proxy class. Its instances stand for one-dimensional arrays that, conceptually, do not exist.

29. *What is cloning?*

An object can carry out copying in two ways i.e. it can set itself to be a copy of another object, or it can return a copy of itself. The latter process is called cloning.

30. *List out some of the OODBMS available.*

- GEMSTONE/OPAL of Gemstone systems.
- ONTOS of Ontos.
- Objectivity of Objectivity inc.
- Versant of Versant object technology.
- Object store of Object Design.
- ARDENT of ARDENT software.
- POET of POET software.

Object Oriented Analysis and Design

1. *What do you mean by analysis and design?*

Analysis:

Basically, it is the process of determining what needs to be done before how it should be done. In order to accomplish this, the developer refers the existing systems and documents. So, simply it is an art of discovery.

Design:

It is the process of adopting/choosing the one among the many, which best accomplishes the users needs. So, simply, it is compromising mechanism.

2. *What are the steps involved in designing?*

Before getting into the design the designer should go through the SRS prepared by the System Analyst. The main tasks of design are Architectural Design and Detailed Design.

In Architectural Design we find what are the main modules in the problem domain.

In Detailed Design we find what should be done within each module.

3. *What are the main underlying concepts of object orientation?*

Objects, messages, class, inheritance and polymorphism are the main concepts of object orientation.

4. *What do u meant by "SBI" of an object?*

SBI stands for State, Behavior and Identity. Since every object has the above three.

➤ *State:*

It is just a value to the attribute of an object at a particular time.

➤ *Behaviour:*

It describes the actions and their reactions of that object.

➤ *Identity:*

An object has an identity that characterizes its own existence. The identity makes it possible to distinguish any object in an unambiguous way, and independently from its state.

5. *Differentiate persistent and non-persistent objects.*

Persistent refers to an object's ability to transcend time or space. A persistent object stores/saves its state in a permanent storage system with out losing the information represented by the object.

A non-persistent object is said to be transient or ephemeral. By default objects are considered as non-persistent.

6. *What do you meant by active and passive objects?*

Active objects are one which instigate an interaction which owns a thread and they are responsible for handling control to other objects. In simple words it can be

referred as *client*.

Passive objects are one, which passively waits for the message to be processed. It waits for another object that requires its services. In simple words it can be referred as *server*.

Diagram:



7. *What is meant by software development method?*

Software development method describes how to model and build software systems in a reliable and reproducible way. To put it simple, methods that are used to represent ones' thinking using graphical notations.

8. *What are models and meta models?*

Model:

It is a complete description of something (i.e. system).

Meta model:

It describes the model elements, syntax and semantics of the notation that allows their manipulation.

9. *What do you meant by static and dynamic modeling?*

Static modeling is used to specify structure of the objects that exist in the problem domain. These are expressed using *class*, *object* and *USECASE diagrams*.

But Dynamic modeling refers representing the object interactions during runtime. It is represented by *sequence*, *activity*, *collaboration* and *statechart diagrams*.

10. *How to represent the interaction between the modeling elements?*

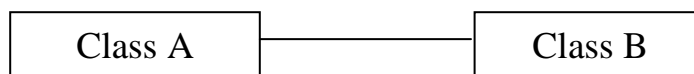
Model element is just a notation to represent (Graphically) the entities that exist in the problem domain. e.g. for modeling element is class notation, object notation etc.

Relationships are used to represent the interaction between the modeling elements.

The following are the Relationships.

- *Association*: Its' just a semantic connection two classes.

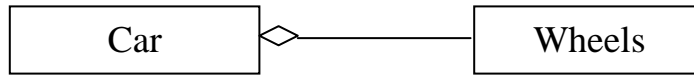
e.g.:



- *Aggregation*: Its' the relationship between two classes which are related in the fashion that *master and slave*. The master takes full rights than the slave. Since the slave works under the master. It is represented as line with diamond in the master area.

Example:

car contains wheels, etc.



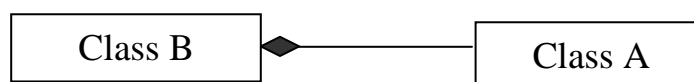
➤ *Containment*: This relationship is applied when the part contained within the whole part, dies when the whole part dies.

It is represented as a darkened diamond at the whole part.

Example:

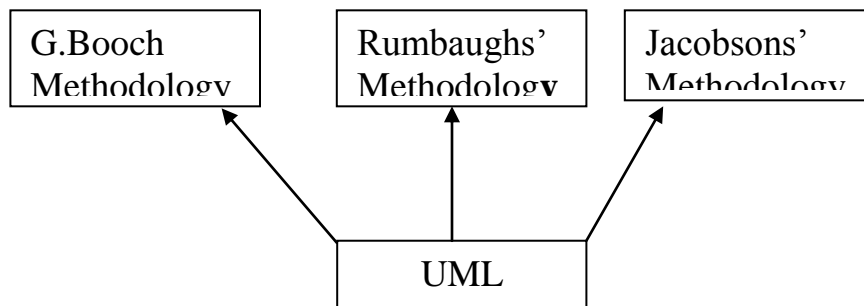
```
class A{
    //some code
};
class B{
    A aa; // an object of class A;
    // some code for class B;
};
```

In the above example we see that an object of class A is instantiated within the class B. So the object of class A dies when the object of class B dies. We can represent it in a diagram like this:



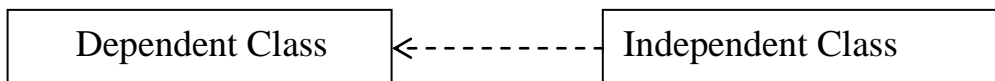
➤ *Generalization*: This relationship used when we want represents a class, which captures the common states of objects of different classes. It is represented as arrow line pointed at the class, which has captured the common states.

The above graph depicts the concept of generalization. Since UML is the class which culminates the properties of the other three (i.e. Booch, Rumbaugh, Jacobson).



➤ *Dependency*: It is the relationship between dependent and independent classes. Any change in the independent class will affect the states of the dependent class.

Diagram:



11. *Why generalization is very strong?*

Even though Generalization satisfies Structural, Interface, Behaviour properties. It is mathematically very strong, as it is Antisymmetric and Transitive.

Antisymmetric:

Employee is a person, but not all persons are employees. Mathematically all As' are B, but all Bs' not A.

Transitive:

$A \Rightarrow B, B \Rightarrow c \text{ then } A \Rightarrow c.$

A. Salesman, B. Employee, C. Person.

Note:

All the other relationships satisfy all the properties like Structural properties, Interface properties, Behaviour properties.

12. *Differentiate Aggregation and containment?*

Aggregation is the relationship between the whole and a part. We can add/subtract some properties in the part (slave) side. It won't affect the whole part.

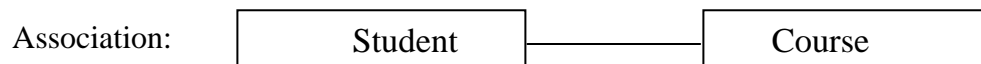
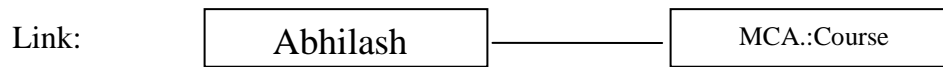
Best example is Car, which contains the wheels and some extra parts. Even though the parts are not there we can call it as car.

But, in the case of containment the whole part is affected when the part within that got affected. The human body is an apt example for this relationship. When the whole body dies the parts (heart etc) are died.

13. *Can link and Association applied interchangeably?*

No, You cannot apply the link and Association interchangeably. Since link is used represent the relationship between the two objects.

But Association is used represent the relationship between the two classes.



14. *List out some of the object-oriented methodologies.*

- Object Oriented Development (OOD) (Booch 1991,1994).
- Object Oriented Analysis and Design (OOA/D) (Coad and Yourdon 1991).
- Object Modelling Techniques (OMT) (Rumbaugh 1991).
- Object Oriented Software Engineering (Objectory) (Jacobson 1992).
- Object Oriented Analysis (OO (Shlaer and Mellor 1992).
- The Fusion Method (Coleman 1991).

15. *What is meant by "method-wars"?*

Before 1994 there were different methodologies like Rumbaugh, Booch, Jacobson, Meyer etc who followed their own notations to model the systems. The

developers were in a dilemma to choose the method which best accomplishes their needs. This particular time-span was called as "method-wars".

16. *Whether unified method and unified modeling language are same or different?*

Unified method is convergence of the Rumbaugh and Booch. Unified modeling lang. is the fusion of Rumbaugh, Booch and Jacobson as well as Bertrand Meyer (whose contribution is "sequence diagram"). Its' the superset of all the methodologies.

17. *Who were the three famous amigos and what was their contribution to the object community?*

The Three amigos namely,

➤ *James Rumbaugh (OMT)*: A veteran in analysis who came up with an idea about the objects and their Relationships (in particular Associations).

➤ *Grady Booch*: A veteran in design who came up with an idea about partitioning of systems into subsystems.

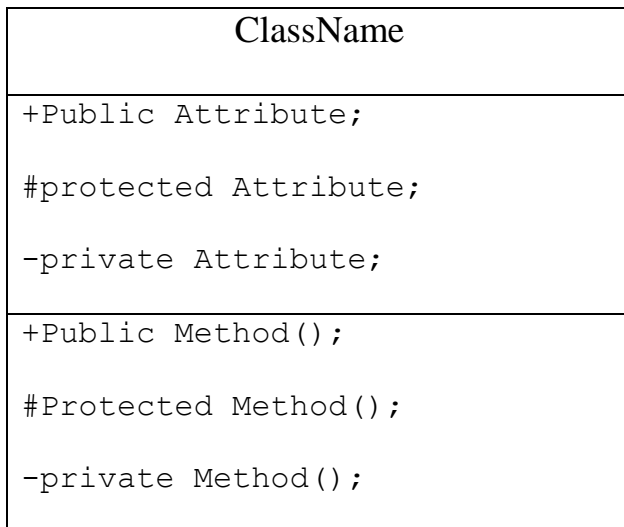
➤ *Ivar Jacobson (Objectory)*: The father of USECASES, who described about the user and system interaction.

17. *Differentiate the class representation of Booch, Rumbaugh and UML?*

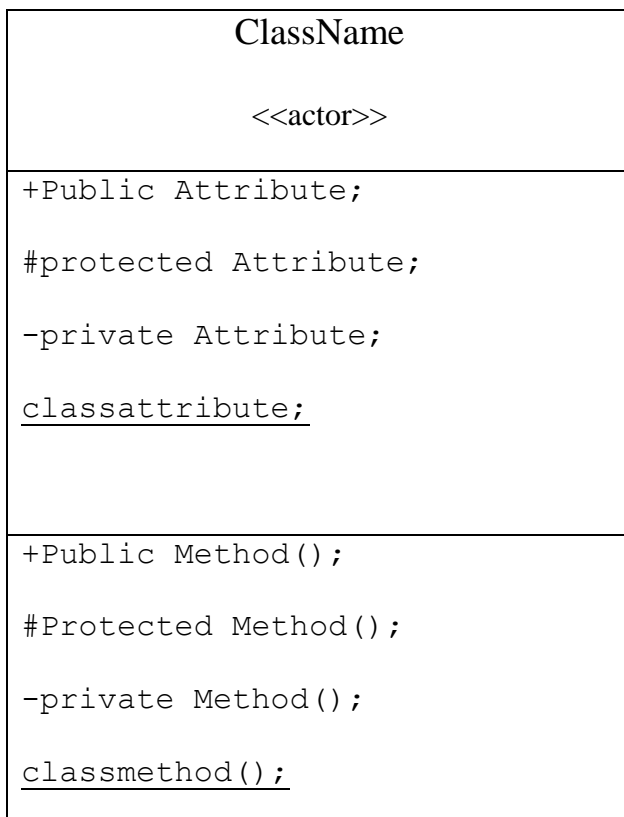
If you look at the class representaiton of Rumbaugh and UML, It is some what similar and both are very easy to draw.

Representation:

OMT



UML.

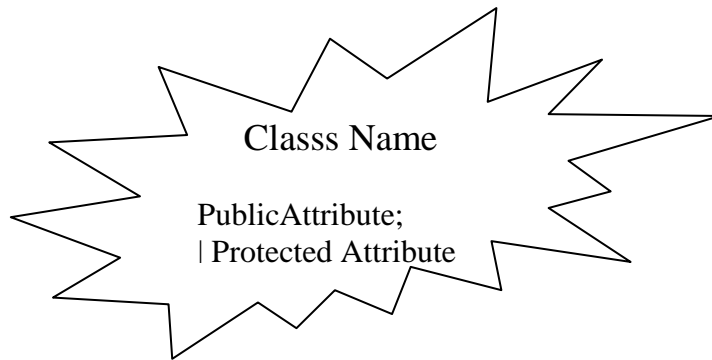




Booch:

In this method classes are represented as "Clouds" which are not very easy to draw as for as the developer's view is concern.

Representation:

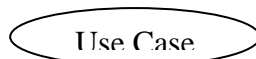


18. *What is an USECASE?why it is needed?*

A Use Case is a description of a set of sequence of actions that a system performs that yields an observable result of value to a particular action.

Simply, in SSAD process \Leftrightarrow In OOAD USECASE. It is represented elliptically.

Representation:



19. *Who is an Actor?*

An Actor is someone or something that must interact with the system. In addition to that an Actor initiates the process (that is USECASE).

It is represented as a stickman like this.

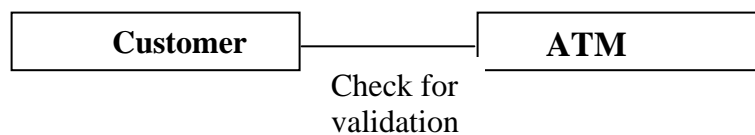
Representation:



20. *What is guard condition?*

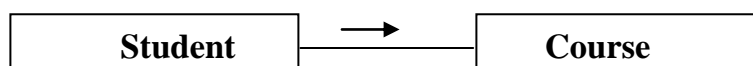
Guard condition is one which acts as a firewall. The access from a particular object can be made only when the particular condition is met.

For Example,

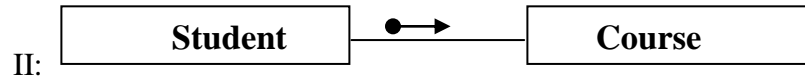


here the object on the customer access the ATM facility only when the guard condition is met.

21. *Differentiate the following notations?*



I:

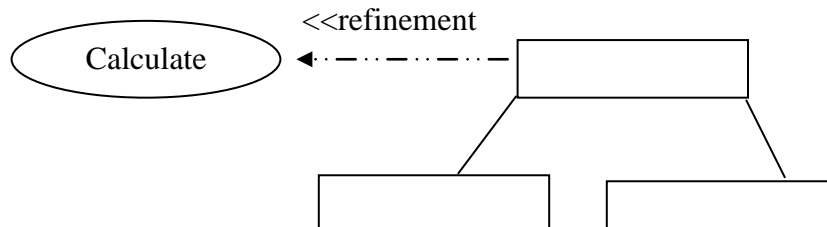


In the above I representation Student Class sends message to Course Class but in the case of second , the data is transfered from student Class to Course Class

22. *USECASE is an implementaion independent notation. How will the designer give the implementaion details of a particular USECASE to the programmer?*

This can be accomplished by specifying the relationship called "refinement" that talkes about the two different abstraction of the same thing.

For example,

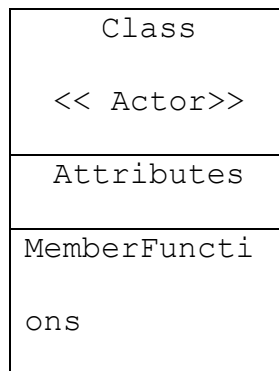


In the above example calculate Pay is an USECASE. It is refined in terms of giving the implementation details. This kind of connection is related by means of “refinement”.

23. *Suppose a class acts an Actor in the problem domain,how can i represent it in the static model?*

In this senario you can use “stereotype”.since stereotype is just a string that gives extra semantic to the particular entity/model element.

It is given with in the << >>.



24. *Why does the function arguments are called as "signatures"?*

The arguments distinguishes functions with the same name (functional polymorphism). The name alone does not necessarily identify a unique function. However, the name and its arguments (signatures) will uniquely identify a function.

In real life we see suppose,in class there are two guys with same name.but they can be easily identified by their signatures.The same concept is applied here.

For example:

```
class person
{
public:
    char getsex();
    void setsex(char);
    void setsex(int);
};
```

In this example we can see that there is a function setsex() with same name but with different signature.

Operating Systems

1. *Explain the concept of Reentrancy.*

It is a useful, memory-saving technique for multiprogrammed timesharing systems. A *Reentrant Procedure* is one in which multiple users can share a single copy of a program during the same period. Reentrancy has 2 key aspects:

- i.) The program code cannot modify itself,
- ii.) The local data for each user process must be stored separately.

Thus, the permanent part is the code, and the temporary part is the pointer back to the calling program and local variables used by that program. Each execution instance is called *activation*. It executes the code in the permanent part, but has its own copy of local variables/parameters. The temporary part associated with each activation is the *activation record*. Generally, the activation record is kept on the stack.

Note: A reentrant procedure *can* be interrupted and called by an interrupting program, and still execute correctly on returning to the procedure.

2. *Explain Belady's Anomaly.*

Also called FIFO anomaly. Usually, on increasing the number of frames allocated to a process' virtual memory, the process execution is faster, because fewer page faults

occur. Sometimes, the reverse happens, i.e., the execution time increases even when more frames are allocated to the process. This is Belady's Anomaly. This is true for certain page reference patterns.

3. *What is a binary semaphore? What is its use?*

A binary semaphore is one, which takes only 0 and 1 as values. They are used to implement mutual exclusion and synchronize concurrent processes.

4. *What is thrashing?*

It is a phenomenon in virtual memory schemes, when the processor spends most of its time swapping pages, rather than executing instructions. This is due to an inordinate number of page faults.

5. *List the Coffman's conditions that lead to a deadlock.*

a) *Mutual Exclusion*: Only one process may use a critical resource at a time.

b) *Hold & Wait*: A process may be allocated some resources while waiting for others.

c) *No Pre-emption*: No resource can be forcibly removed from a process holding it.

d) *Circular Wait*: A closed chain of processes exist such that each process holds at least one resource needed by another process in the chain.

6. *What are short-, long- and medium-term scheduling?*

Long term scheduler determines which programs are admitted to the system for processing. It controls the *degree of multiprogramming*. Once admitted, a job becomes a process.

Medium term scheduling is part of the swapping function. This relates to processes that are in a blocked or suspended state. They are swapped out of main-memory until they are ready to execute. The swapping-in decision is based on memory-management criteria.

Short term scheduler, also known as a *dispatcher* executes most frequently, and makes the finest-grained decision of which process should execute next. This scheduler is invoked whenever an event occurs. It may lead to interruption of one process by preemption.

7. *What are turnaround time and response time?*

Turnaround time is the interval between the submission of a job and its completion. *Response time* is the interval between submission of a request, and the first response to that request.

8. *What are the typical elements of a process image?*

a) *User data*: Modifiable part of user space. May include program data, user stack area, and programs that may be modified.

b) *User program*: The instructions to be executed.

c) *System Stack*: Each process has one or more LIFO stacks associated with it. Used to store parameters and calling addresses for procedure and system calls.

d) *Process Control Block (PCB)*: Info needed by the OS to control processes.

9. *What is the Translation Lookaside Buffer (TLB)?*

In a cached system, the base addresses of the last few referenced pages is maintained in registers called the TLB that aids in faster lookup. TLB contains those page-table entries that have been most recently used. Normally, each virtual memory reference causes 2 physical memory accesses-- one to fetch appropriate page-table entry, and one to fetch the desired data. Using TLB in-between, this is reduced to just one physical memory access in cases of TLB-hit.

10. *What is the resident set and working set of a process?*

Resident set is that portion of the process image that is actually in main-memory at a particular instant. *Working set is that subset of resident set* that is actually needed for execution. (Relate this to the variable-window size method for swapping techniques.)

11. *When is a system in safe state?*

The set of dispatchable processes is in a safe state if there exist at least one temporal order in which all processes can be run to completion without resulting in a deadlock.

12. *What is cycle stealing?*

We encounter cycle stealing in the context of Direct Memory Access (DMA). Either the DMA controller can use the data bus when the CPU does not need it, or it may

force the CPU to temporarily suspend operation. The latter technique is called cycle stealing. Note that cycle stealing can be done only at specific break points in an instruction cycle.

13. *What is meant by arm-stickiness?*

If one or a few processes have a high access rate to data on one track of a storage disk, then they may monopolize the device by repeated requests to that track. This generally happens with most common device scheduling algorithms (LIFO, SSTF, C-SCAN, etc). High-density multi-surface disks are more likely to be affected by this, than the low density ones.

14. *What are the stipulations of C2 level security?*

C2 level security provides for:

1. Discretionary Access Control
2. Identification and Authentication
3. Auditing
4. Resource Reuse

15. *What is busy waiting?*

The repeated execution of a loop of code while waiting for an event to occur is called *busy-waiting*. The CPU is not engaged in any real productive activity during this period, and the process does not progress toward completion.

16. *Explain the popular multiprocessor thread-scheduling strategies.*

Load Sharing: Processes are not assigned to a particular processor. A global queue of threads is maintained. Each processor, when idle, selects a thread from this queue. Note that *load balancing* refers to a scheme where work is allocated to processors on a more permanent basis.

Gang Scheduling: A set of related threads is scheduled to run on a set of processors at the same time, on a 1-to-1 basis. Closely related threads / processes may be scheduled this way to reduce synchronization blocking, and minimize process switching. Group scheduling predated this strategy.

Dedicated processor assignment: Provides implicit scheduling defined by assignment of threads to processors. For the duration of program execution, each program is allocated a set of processors equal in number to the number of threads in the program. Processors are chosen from the available pool.

Dynamic scheduling: The number of thread in a program can be altered during the course of execution.

17. *When does the condition 'rendezvous' arise?*

In message passing, it is the condition in which, both, the sender and receiver are blocked until the message is delivered.

18. *What is a trap and trapdoor?*

Trapdoor is a secret undocumented entry point into a program, used to grant access without normal methods of access authentication. A *trap* is a software interrupt, usually the result of an error condition.

19. *What are local and global page replacements?*

Local replacement means that an incoming page is brought in only to the relevant process' address space. *Global* replacement policy allows any page frame from any process to be replaced. The latter is applicable to variable partitions model only.

20. *Define latency, transfer and seek time with respect to disk I/O.*

Seek time is the time required to move the disk arm to the required track. *Rotational delay* or *latency* is the time to move the required sector to the disk head. Sums of seek time (if any) and the latency is the *access time*, for accessing a particular track in a particular sector. Time taken to actually transfer a span of data is *transfer time*.

21. *Describe the Buddy system of memory allocation.*

Free memory is maintained in linked lists, each of equal sized blocks. Any such block is of size 2^k . When some memory is required by a process, the block size of next higher order is chosen, and broken into two. Note that the two such pieces differ in address only in their k th bit. Such pieces are called *buddies*. When any used block is freed, the OS checks to see if its buddy is also free. If so, it is rejoined, and put into the original free-block linked-list.

22. *What is time stamping?*

It is a technique proposed by Lamport, used to order events in a distributed system without the use of clocks. This scheme is intended to order events consisting of the transmission of messages. Each system 'i' in the network maintains a counter C_i . Every time a system transmits a message, it increments its counter by 1 and attaches the time-stamp T_i to the message. When a message is received, the receiving system 'j' sets its counter C_j to 1 more than the maximum of its current value and the incoming time-stamp T_i . At each site, the ordering of messages is determined by the following rules:

For messages x from site i and messages y from site j , x precedes y if one of the following conditions holds if $T_i < T_j$ or if $T_i = T_j$ and $i < j$.

23. *How are the wait/signal operations for monitor different from those for semaphores?*

If a process in the *monitor* signals and no task is waiting on the condition variable, the signal is lost. So this allows easier program design. Whereas in semaphores, every operation affects the value of the semaphore, so the wait and signal operations should be perfectly balanced in the program.

24. *In the context of memory management, what are placement and replacement algorithms?*

Placement algorithms determine where in the available main-memory to load the incoming process. Common methods are first-fit, next-fit, and best-fit. *Replacement* algorithms are used when memory is full, and one process (or part of a process) needs to be swapped out to accommodate the new incoming process. The replacement algorithm

determines which are the partitions (memory portions occupied by the processes) to be swapped out.

25. *In loading processes into memory, what is the difference between load-time dynamic linking and run-time dynamic linking?*

For *load-time dynamic linking*: Load module to be loaded is read into memory. Any reference to a target external module causes that module to be loaded and the references are updated to a relative address from the start base address of the application module.

With *run-time dynamic loading*: Some of the linking is postponed until actual reference during execution. Then the correct module is loaded and linked.

26. *What are demand- and pre-paging?*

With *demand paging*, a page is brought into the main-memory only when a location on that page is actually referenced during execution. With *prepaging*, pages other than the one demanded by a page fault are brought in. The selection of such pages is done based on common access patterns, especially for secondary memory devices.

27. *What is mounting?*

Mounting is the mechanism by which two different file systems can be combined together. This is one of the services provided by the operating system, which allows the user to work with two different file systems, and some of the secondary devices.

28. *What do you mean by dispatch latency?*

The time taken by the dispatcher to stop one process and start running another process is known as the dispatch latency.

29. *What is multi-processing?*

The ability of an operating system to use more than one CPU in a single computer system. *Symmetrical* multiprocessing refers to the OS's ability to assign tasks dynamically to the next available processor, whereas *asymmetrical* multiprocessing requires that the original program designer choose the processor to use for a given task at the time of writing the program.

30. *What is multitasking?*

Multitasking is a logical extension of multi-programming. This refers to the simultaneous execution of more than one program, by switching between them, in a single computer system.

31. *Define multithreading?*

The concurrent processing of several tasks or threads inside the same program or process. Because several tasks can be processed parallelly and no tasks have to wait for the another to finish its execution.

32. *Define compaction.*

Compaction refers to the mechanism of shuffling the memory portions such that all the free portions of the memory can be aligned (or merged) together in a single large block. OS to overcome the problem of fragmentation, either internal or external, performs this mechanism, frequently. Compaction is possible only if relocation is dynamic and done at run-time, and if relocation is static and done at assembly or load-time compaction is not possible.

33. *What do you mean by FAT (File Allocation Table)?*

A table that indicates the physical location on secondary storage of the space allocated to a file. FAT chains the clusters (group of sectors) to define the contents of the file. FAT allocates clusters to files.

34. *What is a Kernel?*

Kernel is the nucleus or core of the operating system. This represents small part of the code, which is thought to be the entire operating system, it is most intensively used. Generally, the kernel is maintained permanently in main memory, and other portions of the OS are moved to and from the secondary storage (mostly hard disk).

35. *What is memory-mapped I/O?*

Memory-mapped I/O, meaning that the communication between the I/O devices and the processor is done through physical memory locations in the address space. Each I/O device will occupy some locations in the I/O address space. I.e., it will respond when those addresses are placed on the bus. The processor can write those locations to send

commands and information to the I/O device and read those locations to get information and status from the I/O device. Memory-mapped I/O makes it easy to write device drivers in a high-level language as long as the high-level language can load and store from arbitrary addresses.

36. *What are the advantages of threads?*

- Threads provide parallel processing like processes but they have one important advantage over process, they are much more efficient.
- Threads are cheaper to create and destroy because they do not require allocation and de-allocation of a new address space or other process resources.
- It is faster to switch between threads. It will be faster since the memory-mapping does not have to be setup and the memory and address translation caches do not have to be violated.
- Threads are efficient as they share memory. They do not have to use system calls (which are slower because of context switches) to communicate.

37. *What are kernel threads?*

The processes that execute in the Kernel-mode that processes are called kernel threads.

38. *What are the necessary conditions for deadlock to exist?*

- Process claims exclusive control for the Resources allocated to them. (*Mutual exclusion condition*).

- Resources cannot be de-allocated until the process completes they are used for its complete execution. (*No preemption condition*).
- A process can hold one resource and wait for other resources to be allocated. (*Wait for condition*)
- *Circular wait condition*.

39. *What are the strategies for dealing with deadlock?*

- *Prevention*- Place restrictions on resource requests so that deadlock cannot occur.
- *Avoidance*- Plan ahead so that you never get in to a situation where deadlock is inevitable.
- *Recovery*- when deadlock is identified in the system, it recovers from it by removing some of the causes of the deadlock.
- *Detection* – detecting whether the deadlock actually exists and identifies the processes and resources that are involved in the deadlock.

40. *Paging a memory management function, while multiprogramming a processor management function, are the two interdependent?*

Yes.

41. *What is page cannibalizing?*

Page swapping or page replacements are called page cannibalizing.

42. *What has triggered the need for multitasking in PCs?*

- Increased speed and memory capacity of microprocessors together with the support for virtual memory and
- Growth of client server computing

43. *What are the four layers that Windows NT have in order to achieve independence?*

- Hardware abstraction layer
- Kernel
- Subsystems
- System Services.

44. *What is SMP?*

To achieve maximum efficiency and reliability a mode of operation known as symmetric multiprocessing is used. In essence, with SMP any process or threads can be assigned to any processor.

45. *What are the key object oriented concepts used by Windows NT?*

- Encapsulation
- Object class and instance

46. *Is Windows NT a full blown object oriented operating system? Give reasons.*

No Windows NT is not so, because its not implemented in object oriented language and the data structures reside within one executive component and are not represented as objects and it does not support object oriented capabilities .

47. *What is a drawback of MVT?*

It does not have the features like

- ability to support multiple processors
- virtual storage
- source level debugging

48. *What is process spawning?*

When the OS at the explicit request of another process creates a process, this action is called process spawning.

49. *How many jobs can be run concurrently on MVT?*

15 jobs

50. *List out some reasons for process termination.*

- Normal completion
- Time limit exceeded
- Memory unavailable
- Bounds violation
- Protection error
- Arithmetic error
- Time overrun
- I/O failure

- Invalid instruction
- Privileged instruction
- Data misuse
- Operator or OS intervention
- Parent termination.

51. *What are the reasons for process suspension?*

- swapping
- interactive user request
- timing
- parent process request

52. *What is process migration?*

It is the transfer of sufficient amount of the state of process from one machine to the target machine

53. *What is mutant?*

In Windows NT a mutant provides kernel mode or user mode mutual exclusion with the notion of ownership.

54. *What is an idle thread?*

The special thread a dispatcher will execute when no ready thread is found.

55. *What is FtDisk?*

It is a fault tolerance disk driver for Windows NT.

56. *What are the possible threads a thread can have?*

- Ready
- Standby
- Running
- Waiting
- Transition
- Terminated.

57. *What are rings in Windows NT?*

Windows NT uses protection mechanism called rings provides by the process to implement separation between the user mode and kernel mode.

58. *What is Executive in Windows NT?*

In Windows NT, executive refers to the operating system code that runs in kernel mode.

59. *What are the sub-components of I/O manager in Windows NT?*

- Network redirector/ Server
- Cache manager.
- File systems

- Network driver
- Device driver

60. *What are DDks? Name an operating system that includes this feature.*

DDks are device driver kits, which are equivalent to SDKs for writing device drivers. Windows NT includes DDks.

61. *What level of security does Windows NT meet?*

C2 level security.

UNIX Concepts

Section - I - File Management In Unix

1. *What are the logical blocks of the UNIX file system?*

- Boot block
- Super block
- Inode block
- Data block

2. *What is an 'inode'?*

All UNIX files have its description stored in a structure called 'inode'. The inode contains info about the file-size, its location, time of last access, time of last modification, permission and so on. Directories are also represented as files and have an associated inode. In addition to descriptions about the file, the inode contains pointers to the data blocks of the file. If the file is large, inode has indirect pointer to a block of pointers to additional data blocks (this further aggregates for larger files). A block is typically 8k.

Inode consists of the following fields:

- File owner identifier

- File type
- File access permissions
- File access times
- Number of links
- File size
- Location of the file data

3. *How does the inode map to data block of a file?*

Inode has 13 block addresses. The first 10 are direct block addresses and these addresses point to first 10 data blocks in the file.

The 11th address points to a one-level index block.

The 12th address points to a two-level (double in-direction) index block. The 13th address points to a three-level (triple in-direction) index block.

This mapping scheme provides a very large maximum file size with efficient access to large files, still small files are accessed directly in one disk read.

4. *Brief about the directory representation in UNIX*

A UNIX directory is a file containing a correspondence between filenames and inodes. A directory is a special file that the kernel maintains. Only kernel modifies directories, but processes can read directories. The contents of a directory are a list of filename and inode number pairs. When new directories are created, kernel makes two entries named '.' (refers to the directory itself) and '..' (refers to parent directory).

The system call for creating a new directory is `mkdir (pathname, mode)`.

5. *How are devices represented in UNIX?*

All devices are represented by files that are called as *special files*. They are located in '/dev' directory. Thus, device files and other files are named and accessed in the same way.

There are two types of such special files: 'block special files' and 'character special files'. A 'block special file' represents a device with characteristics similar to a disk (data transfer in terms of blocks). A 'character special file' represents a device with characteristics similar to a keyboard (data transfer is by stream of bits in sequential order).

6. *What are the Unix system calls for I/O?*

- open(pathname,flag,mode) - open file
- creat(pathname,mode) - create file
- close(filedes) - close an open file
- read(filedes,buffer,bytes) - read data from an open file
- write(filedes,buffer,bytes) - write data to an open file
- lseek(filedes,offset,from) - position an open file
- dup(filedes) - duplicate an existing file descriptor
- dup2(oldfd,newfd) - duplicate to a desired file descriptor
- fcntl(filedes,cmd,arg) - change properties of an open file
- ioctl(filedes,request,arg) - change the behaviour of an open file

The difference between fcntl and ioctl is that the former is intended for any open

file, while the latter is for device-specific operations.

7. *How do you change File Access Permissions?*

Every file has following attributes:

- owner's user ID (16 bit integer)
- owner's group ID (16 bit integer)
- File access mode word

'r w x - r w x - r w x'

(user permission - group permission - others permission)

r-read, w-write, x-execute.

To change the access mode, we use `chmod(filename,mode)`.

Example 1:

To change mode of myfile to 'rw-rw-r--' (ie. read, write permission for user - read,write permission for group - only read permission for others) we give the args as:

`chmod(myfile,0664)` .

Each operation is represented by discrete values

'r' is 4

'w' is 2

'x' is 1

Therefore, for 'rw' the value is 6(4+2).

Example 2:

To change mode of myfile to 'rwxr--r--' we give the args as:

chmod(myfile,0744).

8. *What are links and symbolic links in UNIX file system?*

A link is a second *name* for a file. Links can be used to assign more than one name to a file, but they cannot be used to assign a directory more than one name or to link filenames on different computers.

Symbolic link 'is' a file that only contains the name of another file. Operation on the symbolic link is directed to the file pointed by the it. Both the limitations of links are eliminated in symbolic links.

Commands for linking files are:

Link	ln filename1 filename2
Symbolic link	ln -s filename1 filename2

9. *What is a FIFO?*

FIFO are otherwise called as 'named pipes'. FIFO (first-in-first-out) is a special file that is said to be 'data transient'. Once data is read from named pipe, it cannot be read again. Also, data can be read only in the order written. It is used in interprocess communication where a process writes to one end of the pipe (producer) and the other reads from the other end (consumer).

10. *How do you create special files like named pipes and device files?*

The system call mknod creates special files in the following sequence:

1. kernel assigns new inode,

2. sets the file type to indicate that the file is a pipe, directory or special file,
3. If it is a device file, it makes the other entries like major, minor device numbers.

For example: If the device is a disk, major device number refers to the disk controller and minor device number refers the disk.

11. *Discuss the mount and unmount system calls*

The privileged mount system call is used to attach a file system to a directory of another file system; the unmount system call detaches a file system. When you mount another file system on to your directory, you are essentially splicing one directory tree onto a branch in another directory tree. The first argument to mount call is the mount point, that is, a directory in the current file naming system. The second argument is the file system to mount to that point. When you insert a cdrom to your Unix system's drive, the file system in the cdrom automatically mounts to /dev/cdrom in your system.

12. *What are surrogate super blocks and surrogate inode tables?*

Whenever we use any file or change its permissions, these changes should be made on the disk; but this can be time consuming. Hence a copy of the super block and an inode table is maintained in the RAM that are called as the surrogate super blocks and inode tables respectively.

The 'sync' command synchronizes the inode table in the memory with the one on the disk by simply overwriting the memory copy on to the disk.

13. *Assuming the block size to be 1KB calculate the maximum size of a file in the Unix file system.*

The first 10 data block pointers can point to 10 data blocks each of size 1 KB .

The 11 th pointer points to a table of block pointers the table has 256 pointers each pointing to data block of size 1 KB. Similarly the 12 th pointer can address

$(256 \times 256\text{KB})$ i.e. 64 MB and the 13 th pointer $(256 \times 64 \text{ MB}) \Rightarrow 16 \text{ GB}$.

Hence the maximum size of the file is 10 KB + 256 KB + 64 MB + 16 GB.

14. *What are the uses of these disk related commands: df, dfspace, du and ulimit?*

\$ df - reports the free as well as used disk space,

\$ dfspace - same as df but is more explanatory,

\$ du - shows the disk space used by a specified file,

\$ ulimit - avoids the user from creating files of very large size.

Section – II

Process Management

1. *Brief about the initial process sequence while the system boots up.*

While booting, special process called the 'swapper' or 'scheduler' is created with the Process-ID 0. The swapper manages memory allocation for processes and influences CPU allocation. The swapper inturn creates 3 children:

➤ the process dispatcher,

➤ vhand and

➤ dbflush

with IDs 1,2 and 3 respectively.

This is done by executing the file /etc/init. Process dispatcher gives birth to the shell. Unix keeps track of all the processes in an internal data structure called the Process Table (listing command is ps -el).

2. *What are various IDs associated with a process?*

Unix identifies each process with an unique integer called ProcessID (PID). The process that executes the request for creation of a process is called the 'parent process' of the newly created process.

Every process is associated with a particular user called the 'owner' who initiates the process and has privileges over the process. The identification for the user is 'UserID'. Process also has 'Effective User ID' that determines the access privileges for accessing resources like files. The system calls used for getting the various IDs are:

getpid() - process id

getppid() - parent process id

getuid() - user id

geteuid() - effective user id

3. *What is the range of values PID can take?*

PID can range from 0 to 32767.

4. *What are the process states in Unix?*

As a process executes it changes state according to its circumstances. Unix

processes have the following states:

Running : The process is either running or it is ready to run .

Waiting : The process is waiting for an event or for a resource.

Stopped : The process has been stopped, usually by receiving a signal.

Zombie : The process is dead but have not been removed from the process table.

5. *What Happens when you execute a program?*

When you execute a program on your UNIX system, the system creates a special environment for that program. This environment contains everything needed for the system to run the program as if no other program were running on the system. Each process has process context, which is everything that is unique about the state of the program you are currently running. Every time you execute a program the UNIX system does a fork, which performs a series of operations to create a process context and then execute your program in that context. The steps include the following:

- Allocate a slot in the process table, a list of currently running programs kept by UNIX.
- Assign a unique process identifier (PID) to the process.
- iCopy the context of the parent, the process that requested the spawning of the new process.
- Return the new PID to the parent process. This enables the parent process to examine or control the process directly.

After the fork is complete, UNIX runs your program.

6. *What Happens when you execute a command?*

When you enter 'ls' command to look at the contents of your current working directory, UNIX does a series of things to create an environment for ls and then run it: The shell has UNIX perform a fork. This creates a new process that the shell will use to run the ls program. The shell has UNIX perform an exec of the ls program. This replaces the shell program and data with the program and data for ls and then starts running that new program. The ls program is loaded into the new process context, replacing the text and data of the shell. The ls program performs its task, listing the contents of the current directory.

7. *What is a zombie process?*

When a program forks and the child finishes before the parent, the kernel still keeps some of its information about the child in case the parent might need it - for example, the parent may need to check the child's exit status. To be able to get this information, the parent calls `wait()`; In the interval between the child terminating and the parent calling `wait()`, the child is said to be a 'zombie' (If you do `ps`, the child will have a 'Z' in its status field to indicate this.)

8. *What is a daemon process?*

A daemon is a process that detaches itself from the terminal and runs, disconnected, in the background, waiting for requests and responding to them. It can also be defined as the background process that does not belong to a terminal session. Many system functions are commonly performed by daemons, including the sendmail daemon,

which handles mail, and the NNTP daemon, which handles USENET news. Many other daemons may exist. Some of the most common daemons are:

- `init`: Takes over the basic running of the system when the kernel has finished the boot process.
- `inetd`: Responsible for starting network services that do not have their own stand-alone daemons. For example, `inetd` usually takes care of incoming `rlogin`, `telnet`, and `ftp` connections.
- `cron`: Responsible for running repetitive tasks on a regular schedule.

Daemons can be roughly classified as system and user daemons.

9. *What is an advantage of executing a process in background?*

The most common reason to put a process in the background is to allow you to do something else interactively without waiting for the process to complete. At the end of the command you add the special background symbol, `&`. This symbol tells your shell to execute the given command in the background.

Example: `cp *.* ../backup&` (cp is for copy)

10. *How do you execute one program from within another?*

The system calls used for low-level process creation are `execlp()` and `execvp()`. The `execlp` call overlays the existing program with the new one, runs that and exits. The original program gets back control only when an error occurs.

`execlp(path,file_name,arguments..);`

//last argument must be NULL

A variant of `execlp` called `execvp` is used when the number of arguments is not known in advance.

```
execvp(path,argument_array);
```

```
//argument array should be terminated by NULL
```

11. *List the system calls used for process management:*

System calls	Description
<code>fork()</code>	To create a new process
<code>exec()</code>	To execute a new program in a process
<code>wait()</code>	To wait until a created process completes its execution
<code>exit()</code>	To exit from a process execution
<code>getpid()</code>	To get a process identifier of the current process
<code>getppid()</code>	To get parent process identifier
<code>nice()</code>	To bias the existing priority of a process
<code>brk()</code>	To increase/decrease the data segment size of a process

12. *Explain `fork()` system call.*

The `fork()` is used to create a new process from an existing process. The new process is called the child process, and the existing process is called the parent. We can tell which is which by checking the return value from `fork()`. The parent gets the child's pid returned to him, but the child gets 0 returned to him.

13. *Predict the output of the following program code*

```
main(){  
    fork();  
    printf("Hello World!");  
}
```

Answer:

Hello World!Hello World!

Explanation:

The fork creates a child that is a duplicate of the parent process. The child begins from the fork(). All the statements after the call to fork() will be executed twice. (once by the parent process and other by child). The statement before fork() is executed only by the parent process.

14. *Predict the output of the following program code*

```
main(){  
    fork(); fork(); fork();  
    printf("Hello World!");  
}
```

Answer:

"Hello World" will be printed 8 times.

Explanation:

2^n times where n is the number of calls to fork()

15. *How can you get/set an environment variable from a program?*

Getting the value of an environment variable is done by using ``getenv()``. Setting the value of an environment variable is done by using ``putenv()``.

16. *How can a parent and child process communicate?*

A parent and child can communicate through any of the normal inter-process communication schemes (pipes, sockets, message queues, shared memory), but also have some special ways to communicate that take advantage of their relationship as a parent and child. One of the most obvious is that the parent can get the exit status of the child.

17. *What is IPC? What are the various schemes available?*

The term IPC (Inter-Process Communication) describes various ways by which different processes running on some operating system communicate between each other. Various schemes available are as follows:

Pipes:

One-way communication scheme through which different process can communicate. The problem is that the two processes should have a common ancestor (parent-child relationship). However this problem was fixed with the introduction of named-pipes (FIFO).

Message Queues :

Message queues can be used between related and unrelated processes running on a machine.

Shared Memory:

This is the fastest of all IPC schemes. The memory to be shared is mapped into

the address space of the processes (that are sharing). The speed achieved is attributed to the fact that there is no kernel involvement. But this scheme needs synchronization.

Various forms of synchronisation are mutexes, condition-variables, read-write locks, record-locks, and semaphores.

18. *Explain 'ps' command and its purpose.*

The `ps` command prints the process status for some or all of the running processes. The information given are the process identification number (PID), the amount of time that the process has taken to execute so far etc.

The options used in this command are

`$ ps -a` :: Lists the processes running for other users.

`$ ps -t` :: Lists the processes running in a particular terminal.

`$ ps -f` :: Lists the processes along with PPID.

`$ ps -e` :: Lists every process running at that instance.

19. *How would you kill a process?*

The `kill` command takes the PID as one argument; this identifies which process to terminate. The PID of a process can be got using 'ps' command.

20. *For some reason, the process with PID 6173 could not be terminated with the command '\$ kill 6173'. What could be the reason and how can you terminate that process?*

The `kill` command when invoked sends a termination signal to the process being killed. Since the signal number is not specified unix assumes the default signal number

which cannot kill certain high priority processes.

In such cases we can use the signal number 9 '\$ kill -9 6173'.

19. *What is a shell?*

A shell is an interactive user interface to services of an operating system, that allows an user to enter commands as character strings or through a graphical user interface. The shell converts them to system calls to the OS or forks off a process to execute the command. Results of the system calls and other information from the OS are presented to the user through an interactive interface. Commonly used shells are sh,csh,ks etc.

21. *Explain about the process priority values.*

Each process is assigned a priority value; higher the value lesser is its priority. The priority value for a process can range from 0 to 39. The default priority value for a process is 20. A user is allowed to increase the value but he cannot decrease it.

22. *What does the command '\$ nice -15 cat emp.dat' do?*

The priority value of the cat emp.dat command is increased form 20 to 35 This will slower the command as higher priority value means lesser priority.

23. *Write a command such that at exactly 5 pm the message "time is 5 pm" appears on the terminal named tty3c.*

\$ at 17:00

```
echo "time is 5 pm" > /dev/tty3c
```

```
ctrl d
```

```
$
```

24. *What does the batch command do?*

The batch command lets the system decide the best time for executing our commands. It may not execute the commands immediately. The batch command will be queued and executed when the system is free.

25. *How is 'crontab' command different from 'at' ?*

The crontab can carry out a submitted job every day for years together without any prompting from the user. The 'at' command is valid only for a day.

Section - III

Memory Management

1. *What is the difference between Swapping and Paging?*

Swapping:

Whole process is moved from the swap device to the main memory for execution. Process size must be less than or equal to the available main memory. It is easier to implementation and overhead to the system. Swapping systems does not handle the memory more flexibly as compared to the paging systems.

Paging:

Only the required memory pages are moved to main memory from the swap device for execution. Process size does not matter. Gives the concept of the virtual memory.

It provides greater flexibility in mapping the virtual address space into the physical memory of the machine. Allows more number of processes to fit in the main memory simultaneously. Allows the greater process size than the available physical memory. Demand paging systems handle the memory more flexibly.

2. *What is the major difference between the Historic Unix and the new BSD release of Unix System V in terms of Memory Management?*

Historic Unix uses Swapping – entire process is transferred to the main memory from the swap device, whereas the Unix System V uses Demand Paging – only the part of the process is moved to the main memory. Historic Unix uses one Swap Device and Unix System V allow multiple Swap Devices.

3. *What is the main goal of the Memory Management?*

- It decides which process should reside in the main memory,
- Manages the parts of the virtual address space of a process which is non-core resident,
- Monitors the available main memory and periodically write the processes into the swap device to provide more processes fit in the main memory simultaneously.

4. *What is a Map?*

A Map is an Array, which contains the addresses of the free space in the swap device that are allocatable resources, and the number of the resource units available there.

Address	Units
1	10,000

This allows First-Fit allocation of contiguous blocks of a resource. Initially the Map contains one entry – address (block offset from the starting of the swap are and the total number of resources.

Kernel treats each unit of Map as a group of disk blocks. On the allocation and freeing of the resources Kernel updates the Map for accurate information.

5. *What is a Region?*

A Region is a continuous area of a process's address space (such as text, data and stack). The kernel in a 'Region Table' that is local to the process maintains region. Regions are sharable among the process.

6. *What are the events done by the Kernel after a process is being swapped out from the main memory?*

When Kernel swaps the process out of the primary memory, it performs the following:

- Kernel decrements the Reference Count of each region of the process. If the reference count becomes zero, swaps the region out of the main memory,

- Kernel allocates the space for the swapping process in the swap device,
- Kernel locks the other swapping process while the current swapping operation is going on,
- The Kernel saves the swap address of the region in the region table.

7. *Is the Process before and after the swap are the same? Give reason.*

Process before swapping is residing in the primary memory in its original form. The regions (text, data and stack) may not be occupied fully by the process, there may be few empty slots in any of the regions and while swapping Kernel do not bother about the empty slots while swapping the process out.

After swapping the process resides in the swap (secondary memory) device. The regions swapped out will be present but only the occupied region slots but not the empty slots that were present before assigning.

While swapping the process once again into the main memory, the Kernel referring to the Process Memory Map, it assigns the main memory accordingly taking care of the empty slots in the regions.

8. *What do you mean by u-area (user area or u-block)?*

This contains the private data that is manipulated only by the Kernel. This is local to the Process, i.e. each process is allocated a u-area.

9. *What are the entities that are swapped out of the main memory while swapping the process out of the main memory?*

All memory space occupied by the process, process's u-area, and Kernel stack are swapped out, theoretically.

Practically, if the process's u-area contains the Address Translation Tables for the process then Kernel implementations do not swap the u-area.

10. *What is Fork swap?*

fork() is a system call to create a child process. When the parent process calls fork() system call, the child process is created and if there is short of memory then the child process is sent to the read-to-run state in the swap device, and return to the user state without swapping the parent process. When the memory will be available the child process will be swapped into the main memory.

11. *What is Expansion swap?*

At the time when any process requires more memory than it is currently allocated, the Kernel performs Expansion swap. To do this Kernel reserves enough space in the swap device. Then the address translation mapping is adjusted for the new virtual address space but the physical memory is not allocated. At last Kernel swaps the process into the assigned space in the swap device. Later when the Kernel swaps the process into the main memory this assigns memory according to the new address translation mapping.

12. *How the Swapper works?*

The swapper is the only process that swaps the processes. The Swapper operates only in the Kernel mode and it does not use System calls instead it uses internal Kernel functions for swapping. It is the archetype of all kernel process.

13. *What are the processes that are not bothered by the swapper? Give Reason.*

- Zombie process: They do not take up any physical memory.
- Processes locked in memories that are updating the region of the process.
- Kernel swaps only the sleeping processes rather than the 'ready-to-run' processes, as they have the higher probability of being scheduled than the sleeping processes.

14. *What are the requirements for a swapper to work?*

The swapper works on the highest scheduling priority. Firstly it will look for any sleeping process, if not found then it will look for the ready-to-run process for swapping. But the major requirement for the swapper to work the ready-to-run process must be core-resident for few seconds before swapping out. And for swapping in the process must have been resided in the swap device for few seconds. If the requirement is not satisfied then the swapper will go into the wait state on that event and it is awakened once in a second by the Kernel.

15. *What are the criteria for choosing a process for swapping into memory from the swap device?*

The resident time of the processes in the swap device, the priority of the processes and the amount of time the processes had been swapped out.

16. *What are the criteria for choosing a process for swapping out of the memory to the swap device?*

- The process's memory resident time,
- Priority of the process and
- The nice value.

17. *What do you mean by nice value?*

Nice value is the value that controls {increments or decrements} the priority of the process. This value that is returned by the nice () system call. The equation for using nice value is:

$$\text{Priority} = (\text{"recent CPU usage"}/\text{constant}) + (\text{base- priority}) + (\text{nice value})$$

Only the administrator can supply the nice value. The nice () system call works for the running process only. Nice value of one process cannot affect the nice value of the other process.

18. *What are conditions on which deadlock can occur while swapping the processes?*

- All processes in the main memory are asleep.
- All 'ready-to-run' processes are swapped out.
- There is no space in the swap device for the new incoming process that are swapped out of the main memory.

- There is no space in the main memory for the new incoming process.

19. *What are conditions for a machine to support Demand Paging?*

- Memory architecture must be based on Pages,
- The machine must support the 'restartable' instructions.

20. *What is 'the principle of locality'?*

It's the nature of the processes that they refer only to the small subset of the total data space of the process. i.e. the process frequently calls the same subroutines or executes the loop instructions.

21. *What is the working set of a process?*

The set of pages that are referred by the process in the last 'n', references, where 'n' is called the *window* of the working set of the process.

22. *What is the window of the working set of a process?*

The window of the working set of a process is the total number in which the process had referred the set of pages in the working set of the process.

23. *What is called a page fault?*

Page fault is referred to the situation when the process addresses a page in the working set of the process but the process fails to locate the page in the working set. And

on a page fault, the kernel updates the working set by reading the page from the secondary device.

24. *What are data structures that are used for Demand Paging?*

Kernel contains 4 data structures for Demand paging. They are,

- Page table entries,
- Disk block descriptors,
- Page frame data table (pfdat),
- Swap-use table.

26. *What are the bits(UNIX System V) that support the demand paging?*

Valid, Reference, Modify, Copy on write, Age. These bits are the part of the page table entry, which includes physical address of the page and protection bits.

Page address	Age	Copy on write	Modify	Reference	Valid	Protection
--------------	-----	---------------	--------	-----------	-------	------------

27. *Difference between the fork() and vfork() system call?*

During the fork() system call the Kernel makes a copy of the parent process's address space and attaches it to the child process.

But the vfork() system call do not makes any copy of the parent's address space, so it is faster than the fork() system call. The child process as a result of the vfork()

system call executes `exec()` system call. The child process from `vfork()` system call executes in the parent's address space (this can overwrite the parent's data and stack) which suspends the parent process until the child process exits.

28. *What is BSS(Block Started by Symbol)?*

A data representation at the machine level, that has initial values when a program starts and tells about how much space the kernel allocates for the un-initialized data. Kernel initializes it to zero at run-time.

29. *What is Page-Stealer process?*

This is the Kernel process that makes rooms for the incoming pages, by swapping the memory pages that are not the part of the working set of a process. Page-Stealer is created by the Kernel at the system initialization and invokes it throughout the lifetime of the system. Kernel locks a region when a process faults on a page in the region, so that page stealer cannot steal the page, which is being faulted in.

30. *Name two paging states for a page in memory?*

The two paging states are:

- The page is aging and is not yet eligible for swapping,
- The page is eligible for swapping but not yet eligible for reassignment to other virtual address space.

31. *What are the phases of swapping a page from the memory?*

- Page stealer finds the page eligible for swapping and places the page number in the list of pages to be swapped.
- Kernel copies the page to a swap device when necessary and clears the *valid* bit in the page table entry, decrements the pfddata reference count, and places the pfddata table entry at the end of the free list if its reference count is 0.

33. *What is page fault? Its types?*

Page fault refers to the situation of not having a page in the main memory when any process references it.

There are two types of page fault :

- Validity fault,
- Protection fault.

34. *What is validity fault?*

If a process referring a page in the main memory whose valid bit is not set, it results in validity fault.

The valid bit is not set for those pages:

- that are outside the virtual address space of a process,
- that are the part of the virtual address space of the process but no physical address is assigned to it.

35. *What do you mean by the protection fault?*

Protection fault refers to the process accessing the pages, which do not have the access permission. A process also incur the protection fault when it attempts to write a page whose *copy on write* bit (UNIX System V) was set during the `fork()` system call.

36. *In what way the Fault Handlers and the Interrupt handlers are different?*

Fault handlers are also an interrupt handler with an exception that the interrupt handlers cannot sleep. Fault handlers sleep in the context of the process that caused the memory fault. The fault refers to the running process and no arbitrary processes are put to sleep.

37. *What does the swapping system do if it identifies the illegal page for swapping?*

If the disk block descriptor does not contain any record of the faulted page, then this causes the attempted memory reference is invalid and the kernel sends a “*Segmentation violation*” signal to the offending process. This happens when the swapping system identifies any invalid memory reference.

38. *What are states that the page can be in, after causing a page fault?*

- On a swap device and not in memory,
- On the free page list in the main memory,
- In an executable file,
- Marked “demand zero”,
- Marked “demand fill”.

39. *In what way the validity fault handler concludes?*

- It sets the valid bit of the page by clearing the modify bit.
- It recalculates the process priority.

40. *At what mode the fault handler executes?*

At the kernel mode.

41. *How the Kernel handles the copy on write bit of a page, when the bit is set?*

In situations like, where the copy on write bit (UNIX System V) of a page is set and that page is shared by more than one process, the Kernel allocates new page and copies the content to the new page and the other processes retain their references to the old page. After copying the Kernel updates the page table entry with the new page number. Then Kernel decrements the reference count of the old pfddata table entry.

In cases like, where the copy on write bit is set and no processes are sharing the page, the Kernel allows the physical page to be reused by the processes. By doing so, it clears the copy on write bit and disassociates the page from its disk copy (if one exists), because other process may share the disk copy. Then it removes the pfddata table entry from the page-queue as the new copy of the virtual page is not on the swap device. It decrements the swap-use count for the page and if count drops to 0, frees the swap space.

42. *For which kind of fault the page is checked first?*

The page is first checked for the validity fault, as soon as it is found that the page is invalid (valid bit is clear), the validity fault handler returns immediately, and the

process incur the validity page fault. Kernel handles the validity fault and the process will incur the protection fault if any one is present.

43. *In what way the protection fault handler concludes?*

After finishing the execution of the fault handler, it sets the *modify* and *protection* bits and clears the *copy on write* bit (all bits as in UNIX System V). It recalculates the process-priority and checks for signals.

44. *How the Kernel handles both the page stealer and the fault handler?*

The page stealer and the fault handler thrash pages because of the shortage of memory. If the sum of the working sets of all processes is greater than the physical memory then the fault handler will usually sleep because it cannot allocate pages for a process. This results in the reduction of the system throughput because Kernel spends too much time in overhead, rearranging the memory in a fast pace.

Computer Networks

1. *Difference between the communication and transmission.*

Transmission is a physical movement of information and concern issues like bit polarity, synchronisation, clock etc.

Communication means the meaning full exchange of information between two communication media.

2. *What are the two types of transmission technology available?*

(i) Broadcast and (ii) point-to-point

3. *What is point-to-point protocol?*

A communications protocol used to connect computers to remote networking services including Internet service providers.

4. *What are the types of Transmission media?*

Signals are usually transmitted over some transmission media that are broadly classified in to two categories.

a) *Guided Media:*

These are those that provide a conduit from one device to another that include twisted-pair, coaxial cable and fiber-optic cable. A signal traveling along any of these media is directed and is contained by the physical limits of the medium. Twisted-pair and coaxial cable use metallic that accept and transport signals in the form of electrical current. Optical fiber is a glass or plastic cable that accepts and transports signals in the form of light.

b) *Unguided Media:*

This is the wireless media that transport electromagnetic waves without using a physical conductor. Signals are broadcast either through air. This is done through radio communication, satellite communication and cellular telephony.

5. *What are the possible ways of data exchange?*

(i) Simplex (ii) Half-duplex (iii) Full-duplex.

6. *What is difference between baseband and broadband transmission?*

In a baseband transmission, the entire bandwidth of the cable is consumed by a single signal. In broadband transmission, signals are sent on multiple frequencies, allowing multiple signals to be sent simultaneously.

7. *What is Protocol Data Unit?*

The data unit in the LLC level is called the protocol data unit (PDU). The PDU contains of four fields a destination service access point (DSAP), a source service access

point (SSAP), a control field and an information field. DSAP, SSAP are addresses used by the LLC to identify the protocol stacks on the receiving and sending machines that are generating and using the data. The control field specifies whether the PDU frame is a information frame (I - frame) or a supervisory frame (S - frame) or a unnumbered frame (U - frame).

8. *What are major types of networks and explain?*

- Server-based network
- Peer-to-peer network

Peer-to-peer network, computers can act as both servers sharing resources and as clients using the resources.

Server-based networks provide centralized control of network resources and rely on server computers to provide security and network administration

9. *What are the important topologies for networks?*

- *BUS topology:*

In this each computer is directly connected to primary network cable in a single line.

Advantages:

Inexpensive, easy to install, simple to understand, easy to extend.

- *STAR topology:*

In this all computers are connected using a central hub.

Advantages:

Can be inexpensive, easy to install and reconfigure and easy to trouble shoot physical problems.

➤ *RING topology:*

In this all computers are connected in loop.

Advantages:

All computers have equal access to network media, installation can be simple, and signal does not degrade as much as in other topologies because each computer regenerates it.

10. *What is passive topology?*

When the computers on the network simply listen and receive the signal, they are referred to as passive because they don't amplify the signal in any way. Example for passive topology - linear bus.

11. *What is mesh network?*

A network in which there are multiple network links between computers to provide multiple paths for data to travel.

12. *What are the different type of networking / internetworking devices?*

Repeater:

Also called a regenerator, it is an electronic device that operates only at physical layer. It receives the signal in the network before it becomes weak, regenerates the original bit pattern and puts the refreshed copy back in to the link.

Bridges:

These operate both in the physical and data link layers of LANs of same type. They divide a larger network in to smaller segments. They contain logic that allow them to keep the traffic for each segment separate and thus are repeaters that relay a frame only the side of the segment containing the intended recipient and control congestion.

Routers:

They relay packets among multiple interconnected networks (i.e. LANs of different type). They operate in the physical, data link and network layers. They contain software that enable them to determine which of the several possible paths is the best for a particular transmission.

Gateways:

They relay packets among networks that have different protocols (e.g. between a LAN and a WAN). They accept a packet formatted for one protocol and convert it to a packet formatted for another protocol before forwarding it. They operate in all seven layers of the OSI model.

13. *How Gateway is different from Routers?*

A gateway operates at the upper levels of the OSI model and translates information between two completely different network architectures or data formats.

14. *What is Brouter?*

Hybrid devices that combine the features of both bridges and routers.

15. *What is subnet?*

A generic term for section of a large networks usually separated by a bridge or router.

16. *What is SAP?*

Series of interface points that allow other computers to communicate with the other layers of network protocol stack.

17. *What do you meant by "triple X" in Networks?*

The function of PAD (Packet Assembler Disassembler) is described in a document known as X.3. The standard protocol has been defined between the terminal and the PAD, called X.28; another standard protocol exists between hte PAD and the network, called X.29. Together, these three recommendations are often called "triple X"

18. *What is frame relay, in which layer it comes?*

Frame relay is a packet switching technology. It will operate in the data link layer.

19. *What is terminal emulation, in which layer it comes?*

Telnet is also called as terminal emulation. It belongs to application layer.

20. *What is Beaconing?*

The process that allows a network to self-repair networks problems. The stations on the network notify the other stations on the ring when they are not receiving the

transmissions. Beaconing is used in Token ring and FDDI networks.

21. *What is redirector?*

Redirector is software that intercepts file or prints I/O requests and translates them into network requests. This comes under presentation layer.

22. *What is NETBIOS and NETBEUI?*

NETBIOS is a programming interface that allows I/O requests to be sent to and received from a remote computer and it hides the networking hardware from applications.

NETBEUI is NetBIOS extended user interface. A transport protocol designed by microsoft and IBM for the use on small subnets.

23. *What is RAID?*

A method for providing fault tolerance by using multiple hard disk drives.

24. *What is cladding?*

A layer of a glass surrounding the center fiber of glass inside a fiber-optic cable.

25. *What is attenuation?*

The degeneration of a signal over distance on a network cable is called attenuation.

26. *What is MAC address?*

The address for a device as it is identified at the Media Access Control (MAC) layer in the network architecture. MAC address is usually stored in ROM on the network adapter card and is unique.

27. *Difference between bit rate and baud rate.*

Bit rate is the number of bits transmitted during one second whereas baud rate refers to the number of signal units per second that are required to represent those bits.

$$\text{baud rate} = \text{bit rate} / N$$

where N is no-of-bits represented by each signal shift.

28. *What is Bandwidth?*

Every line has an upper limit and a lower limit on the frequency of signals it can carry. This limited range is called the bandwidth.

29. *What is Project 802?*

It is a project started by IEEE to set standards that enable intercommunication between equipment from a variety of manufacturers. It is a way for specifying functions of the physical layer, the data link layer and to some extent the network layer to allow for interconnectivity of major LAN protocols.

It consists of the following:

➤ 802.1 is an internetworking standard for compatibility of different LANs and MANs across protocols.

➤ 802.2 Logical link control (LLC) is the upper sublayer of the data link layer which is non-architecture-specific, that is remains the same for all IEEE-defined LANs.

➤ Media access control (MAC) is the lower sublayer of the data link layer that contains some distinct modules each carrying proprietary information specific to the LAN product being used. The modules are Ethernet LAN (802.3), Token ring LAN (802.4), Token bus LAN (802.5).

➤ 802.6 is distributed queue dual bus (DQDB) designed to be used in MANs.

30. *What are the data units at different layers of the TCP / IP protocol suite?*

The data unit created at the application layer is called a message, at the transport layer the data unit created is called either a segment or an user datagram, at the network layer the data unit created is called the datagram, at the data link layer the datagram is encapsulated in to a frame and finally transmitted as signals along the transmission media.

31. *What is ICMP?*

ICMP is Internet Control Message Protocol, a network layer protocol of the TCP/IP suite used by hosts and gateways to send notification of datagram problems back to the sender. It uses the echo test / reply to test whether a destination is reachable and responding. It also handles both control and error messages.

32. *What is difference between ARP and RARP?*

The address resolution protocol (ARP) is used to associate the 32 bit IP address with the 48 bit physical address, used by a host or a router to find the physical address of another host on its network by sending a ARP query packet that includes the IP address of the receiver.

The reverse address resolution protocol (RARP) allows a host to discover its Internet address when it knows only its physical address.

33. *What is the minimum and maximum length of the header in the TCP segment and IP datagram?*

The header should have a minimum length of 20 bytes and can have a maximum length of 60 bytes.

34. *What is the range of addresses in the classes of internet addresses?*

Class A	0.0.0.0	-	127.255.255.255
Class B	128.0.0.0	-	191.255.255.255
Class C	192.0.0.0	-	223.255.255.255
Class D	224.0.0.0	-	239.255.255.255
Class E	240.0.0.0	-	247.255.255.255

35. *What is the difference between TFTP and FTP application layer protocols?*

The Trivial File Transfer Protocol (TFTP) allows a local host to obtain files from a remote host but does not provide reliability or security. It uses the fundamental packet delivery services offered by UDP.

The File Transfer Protocol (FTP) is the standard mechanism provided by TCP / IP for copying a file from one host to another. It uses the services offered by TCP and so is reliable and secure. It establishes two connections (virtual circuits) between the hosts, one for data transfer and another for control information.

36. *Explain 5-4-3 rule?*

In a Ethernet network, between any two points on the network, there can be no more than five network segments or four repeaters, and of those five segments only three of segments can be populated.

37. *What MAU?*

In token Ring , hub is called Multistation Access Unit(MAU).

38. *What is the difference between routable and non- routable protocols?*

Routable protocols can work with a router and can be used to build large networks. Non-Routable protocols are designed to work on small, local networks and cannot be used with a router

39. *Why should you care about the OSI Reference Model?*

It provides a framework for discussing network operations and design.

40. *What is logical link control?*

One of two sublayers of the data link layer of OSI reference model, as defined by the IEEE 802 standard. This sublayer is responsible for maintaining the link between computers when they are sending data across the physical network connection.

41. *What is virtual channel?*

Virtual channel is normally a connection from one source to one destination, although multicast connections are also permitted. The other name for virtual channel is virtual circuit.

42. *What is virtual path?*

Along any transmission path from a given source to a given destination, a group of virtual circuits can be grouped together into what is called path.

43. *What is packet filter?*

Packet filter is a standard router equipped with some extra functionality. The extra functionality allows every incoming or outgoing packet to be inspected. Packets meeting some criterion are forwarded normally. Those that fail the test are dropped.

44. *What is traffic shaping?*

One of the main causes of congestion is that traffic is often busy. If hosts could be made to transmit at a uniform rate, congestion would be less common. Another open loop

method to help manage congestion is forcing the packet to be transmitted at a more predictable rate. This is called traffic shaping.

45. *What is multicast routing?*

Sending a message to a group is called multicasting, and its routing algorithm is called multicast routing.

46. *What is region?*

When hierarchical routing is used, the routers are divided into what we call regions, with each router knowing all the details about how to route packets to destinations within its own region, but knowing nothing about the internal structure of other regions.

47. *What is silly window syndrome?*

It is a problem that can ruin TCP performance. This problem occurs when data are passed to the sending TCP entity in large blocks, but an interactive application on the receiving side reads 1 byte at a time.

48. *What are Digrams and Trigrams?*

The most common two letter combinations are called as digrams. e.g. th, in, er, re and an. The most common three letter combinations are called as trigrams. e.g. the, ing, and, and ion.

49. *What is wide-mouth frog?*

Wide-mouth frog is the simplest known key distribution center (KDC) authentication protocol.

51. *What is Mail Gateway?*

It is a system that performs a protocol translation between different electronic mail delivery protocols.

52. *What is IGP (Interior Gateway Protocol)?*

It is any routing protocol used within an autonomous system.

53. *What is EGP (Exterior Gateway Protocol)?*

It is the protocol the routers in neighboring autonomous systems use to identify the set of networks that can be reached within or via each autonomous system.

54. *What is autonomous system?*

It is a collection of routers under the control of a single administrative authority and that uses a common Interior Gateway Protocol.

55. *What is BGP (Border Gateway Protocol)?*

It is a protocol used to advertise the set of networks that can be reached within an autonomous system. BGP enables this information to be shared with the autonomous system. This is newer than EGP (Exterior Gateway Protocol).

56. *What is Gateway-to-Gateway protocol?*

It is a protocol formerly used to exchange routing information between Internet core routers.

57. *What is NVT (Network Virtual Terminal)?*

It is a set of rules defining a very simple virtual terminal interaction. The NVT is used in the start of a Telnet session.

58. *What is a Multi-homed Host?*

It is a host that has a multiple network interfaces and that requires multiple IP addresses is called as a Multi-homed Host.

59. *What is Kerberos?*

It is an authentication service developed at the Massachusetts Institute of Technology. Kerberos uses encryption to prevent intruders from discovering passwords and gaining unauthorized access to files.

60. *What is OSPF?*

It is an Internet routing protocol that scales well, can route traffic along multiple paths, and uses knowledge of an Internet's topology to make accurate routing decisions.

61. *What is Proxy ARP?*

It is using a router to answer ARP requests. This will be done when the originating host believes that a destination is local, when in fact it lies beyond router.

62. *What is SLIP (Serial Line Interface Protocol)?*

It is a very simple protocol used for transmission of IP datagrams across a serial line.

63. *What is RIP (Routing Information Protocol)?*

It is a simple protocol used to exchange information between the routers.

64. *What is source route?*

It is a sequence of IP addresses identifying the route a datagram must follow. A source route may optionally be included in an IP datagram header.

65. *What is a TCP connection table? How is it used?*

TCP uses a connection table to keep track of all the active connections. Each existing connection has an entry in the table that shows information about the end-to-end connection.

66. *What are the advantages and disadvantages of the three types of routing tables?*

The three types of routing tables are fixed, dynamic, and fixed central. The fixed table must be manually modified every time there is a change. A dynamic table changes its information based on network traffic, reducing the amount of manual maintenance. A

fixed central table lets a manager modify only one table, which is then read by other devices. The fixed central table reduces the need to update each machine's table, as with the fixed table. Usually a dynamic table causes the fewest problems for a network administrator, although the table's contents can change without the administrator being aware of the change.

67. *What is the HELLO protocol used for?*

The HELLO protocol uses time instead of distance to determine optimal routing. It is an alternative to the Routing Information Protocol.

68. *What is the difference between interior and exterior neighbor gateways?*

Interior gateways connect LANs of one organization, whereas exterior gateways connect the organization to the outside world.

69. *What protocol is used by DNS name servers?*

DNS uses UDP for communication between servers. It is a better choice than TCP because of the improved speed a connectionless protocol offers. Of course, transmission reliability suffers with UDP.

70. *What is a DNS resource record?*

A resource record is an entry in a name server's database. There are several types of resource records used, including name-to-address resolution information. Resource records are maintained as ASCII files.

71. *BOOTP helps a diskless workstation boot. How does it get a message to the network looking for its IP address and the location of its operating system boot files?*

BOOTP sends a UDP message with a subnetwork broadcast address and waits for a reply from a server that gives it the IP address. The same message might contain the name of the machine that has the boot files on it. If the boot image location is not specified, the workstation sends another UDP message to query the server.

72. *What is the Network Time Protocol? Why is it used?*

The Network Time Protocol is designed specifically to ensure that all internal timers are synchronized and have similar settings.

73. *What is External Data Representation?*

External Data Representation is a method of encoding data within an RPC message, used to ensure that the data is not system-dependent.

74. *What does the Mount protocol do?*

The Mount protocol returns a file handle and the name of the file system in which a requested file resides. The message is sent to the client from the server after reception of a client's request.

75. *What is REX? What advantage does REX offer other similar utilities?*

The Remote Execution Service (REX) is designed to enable users to run commands on other machines without logging in and without the overhead of larger utilities like Telnet. Because REX preserves the full shell environment, it is better than other remote procedures.

76. *What is a pseudo tty?*

A pseudo tty or false terminal enables external machines to connect through Telnet or rlogin. Without a pseudo tty, no connection can take place.

77. *What is anonymous FTP and why would you use it?*

Anonymous FTP enables users to connect to a host without using a valid login and password. Usually, anonymous FTP uses a login called anonymous or guest, with the password usually requesting the user's ID for tracking purposes only. Anonymous FTP is used to enable a large number of users to access files on the host without having to go to the trouble of setting up logins for them all. Anonymous FTP systems usually have strict controls over the areas an anonymous user can access.

78. *What is a Management Information Base (MIB)?*

A Management Information Base is part of every SNMP-managed device. Each SNMP agent has the MIB database that contains information about the device's status, its performance, connections, and configuration. The MIB is queried by SNMP.

79. *Explain the function of Transmission Control Block?*

A TCB is a complex data structure that contains a considerable amount of information about each connection.

80. *What is the difference between an unspecified passive open and a fully specified passive open?*

An unspecified passive open has the server waiting for a connection request from a client. A fully specified passive open has the server waiting for a connection from a specific client.

81. *Define the following:*

Burst Mode—A transmission mode where data is transmitted in bursts rather than in continuous streams.

Contention—A condition occurring in some LANs where the Media Access Control (MAC) sublayer allows more than one node to transmit at the same time, risking collisions.

Daemon—A UNIX process that operates continuously and unattended to perform a service. TCP/IP uses several daemons to establish communications processes and provide server facilities.

Distributed File Service (DFS)—An Open Software Foundation (OSF) fileserver technology sometimes used with TCP/IP.

Emulation—A program that simulates another device. For example, a 3270 emulator emulates an IBM 3270 terminal, sending the same codes as the real device would.

Enterprise Network—A generic term usually referring to a Wide Area Network providing services to all of a corporation's sites.

Integrated Service Digital Network (ISDN)—A set of standards for integrating multiple services (voice, data, video, and so on).

IP Datagram—The basic unit of information passed through a TCP/IP network. The datagram header contains source and destination IP addresses.

IS-IS (Intermediate System to Intermediate System Protocol)—A routing protocol that performs routing functions with IP and OSI data.

Multi-homed Host—A device attached to two or more networks.

Network File System (NFS)—A protocol developed by Sun Microsystems that enables clients to mount remote directories onto their own local file system

Point-to-Point—Transmission directly between two points without any intervening devices

PING (Packet Internet Gropher)—A utility program used to test a system's TCP/IP software by sending an ICMP echo request and then waiting for a response

Protocol Interpreter (PI)—A process that carries out FTP functions. FTP uses one Protocol Interpreter for the server and another one for the user.

Proxy—A mechanism whereby one system functions for another when responding to protocol requests.

Requests for Comment (RFCs)—Documents containing specifications for TCP/IP protocols. RFCs are also used to propose new protocols. RFCs are available from the Network Information Center (NIC).

Round Trip Time—The time for a TCP segment to be sent and its acknowledgment received.

Switched Connection—A data link connection that is established on demand (like a telephone call).

Transmission Control Block (TCB)—A data structure that holds information about TCP and UDP connections.

Terminal Server—A network device that provides physical access for dumb terminals, usually using an abbreviated TCP/IP protocol to enable a dumb terminal to remotely log on.

X.400—A protocol defining standards for electronic mail in an open network.

82. *What are 10Base2, 10Base5 and 10BaseT Ethernet LANs?*

10Base2—An Ethernet term meaning a maximum transfer rate of 10 Megabits per second that uses baseband signaling, with a contiguous cable segment length of 100 meters and a maximum of 2 segments.

10Base5—An Ethernet term meaning a maximum transfer rate of 10 Megabits per second that uses baseband signaling, with 5 continuous segments not exceeding 100 meters per segment.

10BaseT—An Ethernet term meaning a maximum transfer rate of 10 Megabits per second that uses baseband signaling and twisted pair cabling.

Client/Server Computing

1. *What are the five major technologies that can be used to create Client/Server applications?*

- Database Servers
- TP Monitors
- Groupware
- Distributed Objects
- Intranets.

2. *What is Client/Server?*

Clients and Servers are separate logical entities that work together over a network to accomplish a task. Many systems with very different architectures that are connected together are also called Client/Server.

3. *What are the characteristics of Client/Server?*

- Service
- Shared resources
- Asymmetrical protocols
- Transparency of location
- Mix-and-match
- Message based exchanges
- Encapsulation of services
- Scalability
- Integrity

Client/Server computing is the ultimate "Open platform". It gives the freedom to *mix-and-match* components of almost any level. Clients and servers are *loosely coupled systems* that interact through a message-passing mechanism.

4. *What is meant by Asymmetrical protocols?*

There is a many-to-one relationship between clients and server. Clients always initiate the dialog by requesting a service. Servers are passively awaiting for requests from clients.

5. *What is meant by Mix-and-match?*

The ideal Client/Server software is independent of hardware and Operating System software platform. It should be able to mix-and-match client and server platforms.

6. *What is meant by Horizontal scaling and Vertical scaling?*

Horizontal scaling means adding or removing client workstations with only a slight performance impact.

Vertical scaling means migrating to a larger and faster server machine or multiservers.

7. *What are the types of Servers?*

- File servers
- Database servers
- Transaction servers
- Groupware servers
- Object servers
- Web servers

8. *What are called Transactions?*

The grouped SQL statements are called Transactions (or) A transaction is a collection of actions embused with ACID properties.

9. *What is ACID property?*

ACID is a term coined by Andrew Reuter in 1983, which stands for Atomicity, Consistence, Isolation and Durability.

10. *What are the two types of OLTP?*

- TP lite, based on stored procedures.
- TP heavy, based on the TP monitors.

11. *What is Groupware server?*

Groupware addresses the management of semi-structured information such as text, image, mail, bulletin boards and the flow of work. These Client/Server systems have people in direct contact with other people.

12. *What is a File server?*

File servers are useful for sharing files across a network. With a file server, the client passes requests for file records over network to file server.

13. *What is a Database Server?*

With a database server, the client passes SQL requests as messages to the database server. The results of each SQL command are returned over the network. The server uses its own processing power to find the request data instead of passing all the records back to the client and then getting it find its own data. The result is a much more efficient use of distributed processing power. It is also known as *SQL engine*.

14. *What is a Transaction server?*

With a transaction server, the client invokes remote procedures that reside on the server with an SQL database engine. These remote procedures on the server execute a

group of SQL statements. The network exchange consists of a single request/reply message. The SQL statements either all succeed or fail as a unit.

15. *What is an Object server?*

With an object server, the Client/Server application is written as a set of communicating objects. Client objects communicate with server objects using an *Object Request Broker (ORB)*. The client invokes a method on a remote object. The ORB locates an instance of that object server class, invokes the requested method and returns the results to the client object. Server objects must provide support for concurrency and sharing. The ORB brings it all together.

16. *What is a Web server?*

This new model of Client/Server consists of thin, portable, "universal" clients that talk to superfat servers. In the simplest form, a web server returns documents when clients ask for them by name. The clients and server communicate using an RPC-like protocol called HTTP.

17. *What are called Fat clients and Fat servers?*

If the bulk of the application runs on the Client side, then it is Fat clients. It is used for decision support and personal software.

If the bulk of the application runs on the Server side, then it is Fat servers. It tries to minimize network interchanges by creating more abstract levels of services.

18. *What is meant by Middleware?*

Middleware is a distributed software needed to support interaction between clients and servers. In short, it is the software that is in the middle of the Client/Server systems and it acts as a bridge between the clients and servers. It starts with the API set on the client side that is used to invoke a service and it covers the transmission of the request over the network and the resulting response.

It neither includes the software that provides the actual service - that is in the servers domain nor the user interface or the application login - that's in clients domain.

19. *What are the two broad classes of middleware?*

- General middleware
- Service-specific middleware

20. *What are General Middleware?*

It includes the communication stacks, distributed directories, authentication services, network time, RPC, Queuing services along with the network OS extensions such as the distributed file and print services.

21. *What are Service-specific middleware?*

It is needed to accomplish a particular Client/Server type of services which includes:

- Database specific middleware
- OLTP specific middleware

- Groupware specific middleware
- Object specific middleware
- Internet specific middleware and
- System management specific middleware.

22. *What are the most typical functional units of the Client/Server applications?*

- User interface
- Business Logic and
- Shared data.

23. *What is meant by 2-Tier architecture?*

In 2-tier Client/Server systems, the application logic is either burried inside the user interface on the client or within the database on the server.

Example: File servers and Database servers with stored procedures.

24. *What is meant by 3-Tier architecture?*

In 3-tier Client/Server systems, the application logic (or process) lives in the middle tier and it is separated from the data and the user interface. In theory, the 3-tier Client/Server systems are more scalable, robust and flexible.

Example: TP monitor, Web.

25. *What are the building blocks of Client/Server?*

- The client

- The server and
- Middleware

26. *Explain the building blocks of Client/Server?*

- The client side building block runs the client side of the application.
- The server side building block runs the server side of the application.
- The middleware building block runs on both the client and server sides of an

application. It is broken into three categories

1. Transport stack
2. Network OS
3. Service-specific middleware.

27. *What are the functions of the typical server program?*

- It waits for client-initiated requests.
- Executes many requests at the same time.
- Takes care of VIP clients first.
- Initiates and runs background task activity.
- Keeps running.
- Grown bigger and faster.

28. *What are the services provided by the Operating System?*

- Base services - It is a part of the standard OS

- Extended services - These are add-on modular software components that are layered on top of base service

29. *What are all the Base services provided by the OS?*

- Task preemption
- Task priority
- Semaphores
- Interprocess communications (IPC)
- Local/Remote Interprocess communication
- Threads
- Intertask protection
- Multiuser
- High performance file system
- Efficient memory management and
- Dynamically linked Run-time extensions

30. *What are all the Extended services provided by the OS?*

- Ubiquitous communications
- Network OS extension
- Binary large objects (BLOBs)
- Global directories and Network yellow pages
- Authentication and Authorization services
- System management

- Network time
- Database and transaction services
- Internet services
- Object- oriented services

31. *What are Super servers?*

These are fully-loaded machines which includes multiprocessors, high-speed disk arrays for intensive I/O and fault tolerant features.

32. *What are the two forms of Multi-processing?*

- Asymmetric Multi-processing
- Symmetric Multi-processing

33. *What is meant by Asymmetric Multiprocessing (AMP)?*

It imposes hierarchy and a division of labour among processors. Only one designated processor, the master, controls (in a tightly coupled arrangement) slave processors dedicated to specific functions.

34. *What is meant by Symmetric Multiprocessing (SMP)?*

It treats all processors as equal. Any processor can do the work of any other processor. Applications are divided into threads that can run concurrently on any available processor. Any processor in the pool can run the OS kernel and execute user-written threads.

35. *What are the Classification of clients?*

- Non-GUI clients - Two types are
 1. Non-GUI clients that do not need multi-tasking
(Example: Automatic Teller Machines (ATM), Cell phone)
 2. Non-GUI clients that need multi-tasking
(Example: ROBOTS)
- GUI clients
- OOUI clients

36. *What are called Non-GUI clients, GUI Clients and OOUI Clients?*

Non-GUI Client: These are applications, generate server requests with a minimal amount of human interaction.

GUI Clients: These are applicatoins, where occassional requests to the server result from a human interacting with a GUI
(Example: Windows 3.x, NT 3.5)

OOUI clients : These are applications, which are highly-iconic, object-oriented user interface that provides seamless access to information in very visual formats.
(Example: MAC OS, Windows 95, NT 4.0)

37. *What is meant by Transparency?*

Transparency really means hiding the network and its servers from the users and even the application programmers.

38. *What are the types of Transparencies?*

The types of transparencies the NOS middleware is expected to provide are

- Location transparency
- Namespace transparency
- Logon transparency
- Replication transparency
- Local/Remote access transparency
- Distributed time transparency
- Failure transparency and
- Administration transparency.

39. *What is Remote Procedure Call (RPC)?*

RPC hides the intricacies of the network by using the ordinary procedure call mechanism familiar to every programmer. A client process calls a function on a remote server and suspends itself until it gets back the results. Parameters are passed like in any ordinary procedure. The RPC, like an ordinary procedure, is synchronous. The process that issues the call waits until it gets the results. Under the covers, the RPC run-time software collects values for the parameters, forms a message, and sends it to the remote server. The server receives the request, unpack the parameters, calls the procedures, and sends the reply back to the client. It is a *telephone-like* metaphor.

40. *What is Message Oriented Middleware (MOM)?*

MOM allows general purpose messages to be exchanged in a Client/Server system using message queues. Applications communicate over networks by simply putting messages in the queues and getting messages from queues. It typically provides a very simple high level APIs to its services.

MOM's messaging and queuing allow clients and servers to communicate across a network without being linked by a private, dedicated, logical connection. The clients and server can run at different times. It is a *post-office* like metaphor.

41. *What is Structured Query Language (SQL)?*

SQL is a powerful set-oriented language which was developed by IBM research for the databases that adhere to the relational model. It consists of a short list of powerful, yet highly flexible, commands that can be used to manipulate information collected in tables. Through SQL, we can manipulate and control sets of records at a time.

42. *What are the roles of SQL?*

- SQL is an interactive query language for ad hoc database queries.
- SQL is a database programming language.
- SQL is a data definition and data administration language.
- SQL is the language of networked database servers
- SQL helps protect the data in a multi-user networked environment.

Because of these multifaceted roles it plays, physicists might call SQL as "*The grand unified theory of database*".

43. *What are the three types of SQL database server architecture?*

- Process-per-client Architecture. (Example: Oracle 6, Informix)
- Multithreaded Architecture. (Example: Sybase, SQL server)
- Hybrid Architecture (Example: Oracle 7)

44. *What are Stored procedures?*

A stored procedure is a named collection of SQL statements and procedural logic that is compiled, verified and stored in a server database. It is typically treated like any other database object. Stored procedures accept input parameters so that a single procedure can be used over the network by multiple clients using different input data. A single remote message triggers the execution of a collection of stored SQL statements. The result is a reduction of network traffic and better performance.

45. *What are Triggers and Rules?*

Triggers are special user defined actions usually in the form of stored procedures, that are automatically invoked by the server based on data related events. It can perform complex actions and can use the full power of procedural languages.

A rule is a special type of trigger that is used to perform simple checks on data.

46. *What is the difference between trigger and rule?*

The triggers are called implicitly by database generated events, while stored procedures are called explicitly by client applications.

47. *What is OLTP?*

In the transaction server, the client component usually includes GUI and the server components usually consists of SQL transactions against a database. These applications are called OLTP (*Online Transaction Processing*)

OLTP Applications typically,

- Receive a fixed set of inputs from remote clients.
- Perform multiple pre-compiled SQL comments against a local database.
- Commit the work and
- Return a fixed set of results.

48. *What is a TP Monitor?*

There is no commonly accepted definition for a TP monitor. According to *Jeri Edwards'* a TP Monitor is "an OS for transaction processing".

TP Monitor does mainly two things extremely well. They are

- Process management and
- Transaction management

They were originally introduced to run classes of applications that could service hundreds and sometimes thousands of clients. TP Monitors provide an OS - on top of existing OS - that connects in real time these thousands of humans with a pool of shared server processes.

49. *What is Load balancing?*

If the number of incoming clients requests exceeds the number of processes in a server class, the TP Monitor may dynamically start new ones and this is called Load balancing.

50. *What are the main components of Transaction-based Systems?*

- Resource Manager
- Transaction Manager and
- Application Program

51. *List out the benefits obtained by using the Client/Server oriented TP Monitors*

- Client/Server applications development framework
- Firewalls of protection
- High availability
- Load balancing
- MOM integration
- Scalability of functions
- Reduced system cost.

52. *What are TP-Lite and TP-Heavy Monitors?*

TP-Lite is simply the integration of TP Monitor functions in the database engines.

TP-Heavy are TP Monitors which supports the Client/Server architecture and allow PC to initiate some very complex multiserver transaction from the desktop.