

Q1) What kind of datastructure is used in your mobile phone?

Ans. It depends upon the general implementation of file system by OS. But general most common datastructure used here is "Tree". Often referred as a directory tree system or file system tree. Within each node of tree the meta data of files can be stored. Such as size, Name etc.

```
Class Directory
    string Name
    List<Directory> Directories
    List<string> Files
```

Q2. What is memory leak? And How to avoid it?

Ans Memory leak occurs when programmer creates a **memory in a heap** and forget to delete it. Consequences of memory is that it reduces the performance of the computer. Since there is not much memory available for other processes.

Example How memory leak occurs.

```
void f() {
```

```
    int *ptr = (int *)malloc(sizeof(int));
```

```
    // DO some work
```

```
    return;
```

```
3
```

Here we are allocating memory from heap but we are not free it after the completion of the function.

To avoid a memory leak we can use destructor i.e which after process is complete we can free memory or without destructor is also possible shown in this example

```
void f()
```

```
{
```

```
    int *ptr = (int *)malloc(sizeof(int));
```

```
    // DO some work
```

```
    free(ptr)
```

```
    return
```

```
3
```

Q3, Explain KMP Algorithm.

Ans In Computer Science pattern matching between two strings is very important.

Naive algorithm work $O(m(n-m+1))$ where each string compare back and forth

The KMP matching algorithm uses degenerative property (pattern having the same sub-pattern appearing more than once in the (pattern) of the pattern and improve worst case complexity to $O(m+n)$.

The basic idea behind KMP algorithm is whenever we detect a mismatch (after some matches). we already know some of the character in the text of the next windows. we take advantage of this information to avoid matching the character that we know will anyway match.

Q4) What is difference between dynamic programming and divide and conquer?

Ans

P.T.O

Divide and Conquer

Subproblems are solved independently, and finally all solutions are arrived at the final answer.

Divide and Conquer is slow than dynamic programming

Maximize time for execution

Recursive technique is used in Divide & Conquer

Top-down approach is used in divide and conquer

Results are not stored during computing

Dynamic Programming

Dynamic programming considers large numbers of decision sequences and all the overlapping substances.

Reduces the amount of time spent on execution by consuming less time.

Non-Recursive technique is used in Dynamic programming

Dynamic programming Solution tree bottom up approach is used

Results are stored during execution

Q5) Disadvantage of Using Stack.

- 1> Limited Capacity
- 2> No random access
- 3> Memory management
- 4> not suitable for certain application
- 5> Stack overflow and Underflow
- 6> Recursive function call limitation.

Q6) What is View in DBMS?

Views in SQL are kind of virtual tables. A view also has rows and columns as they are in a real table.

We can create view by select fields from multiple tables in same database. Views can have all rows of tables or certain rows based on certain condition.

We can create view using "CREATE VIEW"

Example

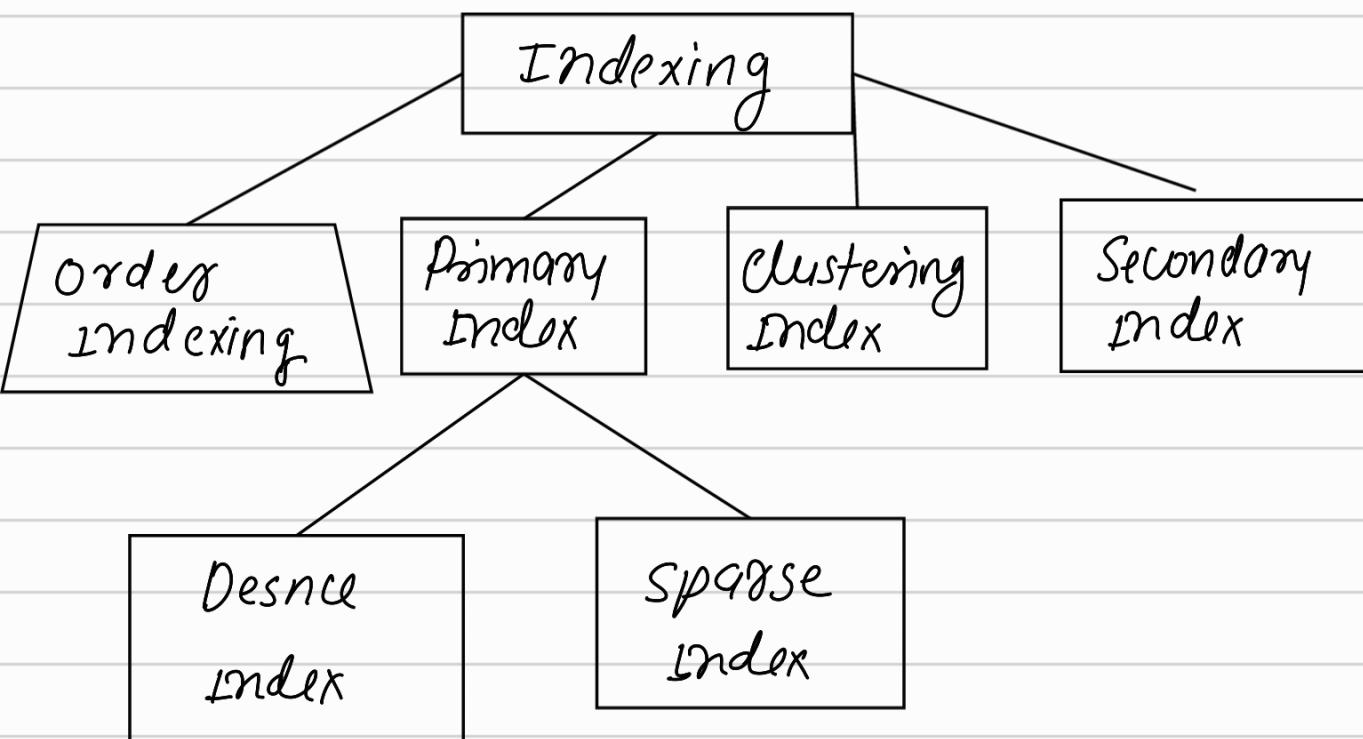
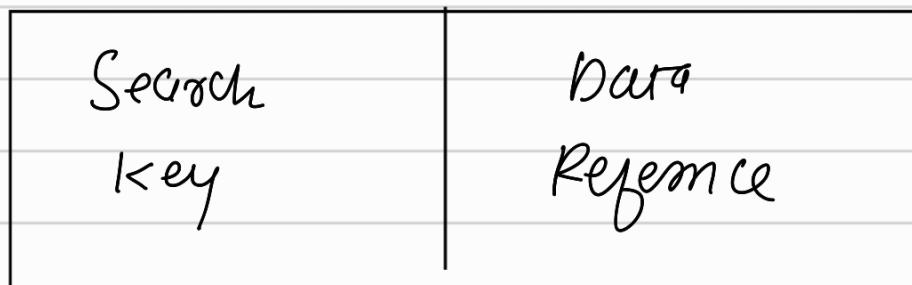
CREATE VIEW view-name AS

SELECT Column1, Column2.....
FROM table-name

WHERE condition;

Q7> What is Indexing?

- * Indexing is used to basically Optimise the performance of database by minimizing disk access requirement when a query is processed.
- * Index is a type of datastruct that is used to access the data in database quickly.

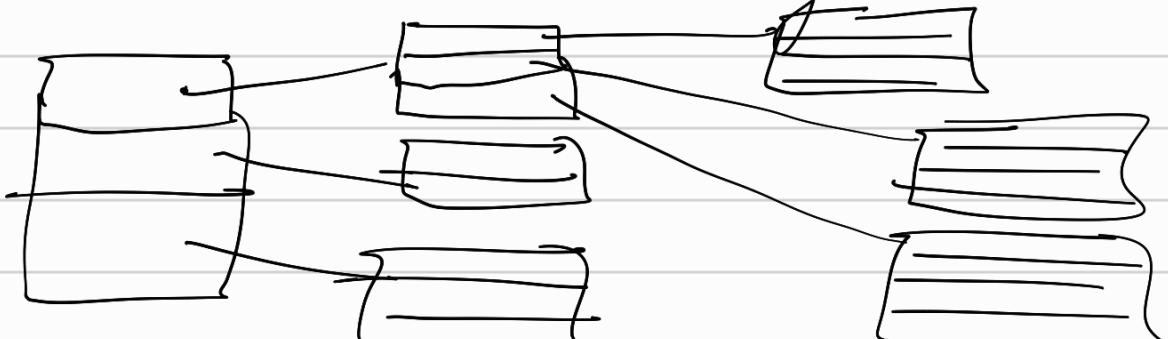


c) Primary Index Vs Secondary Index Vs clustering Index , multi level Index.

Ans

Primary Index	Secondary Index
Index on a set of field that includes the unique primary key and guaranteed not to contain duplicate.	Index that is not a Primary Index and may have duplicate.
Required row on data block to be ordered on the Index key.	Does not not have impact on how the rows are actually organised in data blocks.
There is only one primary Index.	The can be multiple secondary Index.

clustering basically multi level index



Q) Sparse Index Vs Dense Index

Ans Dense Index: In dense Index, there is an Index record for every Search key value in database. This makes searching faster but requires more space to store Index record itself. Index records contains Search key value and pointer to the actual record on the disk.

Sparse Index: In Sparse index, Index

Record are not created for every Search key. An Index record here contains a Search key and an actual pointer to the data on the disk. Basically we first search through index if we found element we go to local store where Index is pointing. Otherwise we go towards the key value list and start Sequential search.

(Q) Difference between lossless and lossy Decomposition.

P.T.O $\downarrow \rightarrow$

Lossless

- * The decomposition $R_1, R_2, R_3 \dots R_n$ is said to be lossless if three natural join of results in relation R .

$$R_1 \bowtie R_2 \bowtie \dots \bowtie R_n = R$$
- * NO loss of information

* The common attribute of the sub relation is super key of any one of the relations

Lossy

- * The decomposition $R_1, R_2, R_3 \dots R_n$ is said to be lossy if true natural join results in relation R with additional tuples.

$$R_1 \bowtie R_2 \bowtie \dots \bowtie R_n \subset R$$
- * Loss of Information

* The common attribute of the sub relation is not the superkey of any one of the relations.

Q) What is Pagefault and why it Occurs.

Ans when a program tries to access the page from the memory which does not exist in the main memory such situation is called Pagefault.

Basically Pagefault occurs if the comⁿ page required by program is not available in main memory so the CPU has to take page from the secondary memory and load back.

c) What is Virtual memory?

Ans Virtual memory is a Storage Allocation schema in which Secondary memory can be addressed as though it is a part of main memory. This technique is implemented both by Software and hardware. It maps memory addresses used by program, called Virtual address, into physical address in computer.

Q) What is Demand Paging?

Ans The process of loading the page into memory on demand (whenever a page fault occurs) is known as demand paging.

There are steps in Demand Paging team i-l

o) Explain Semaphores.

Ans

Semaphores are normal Variable used to coordinate activities of multiple processes in a computer system. They are used to enforce mutual exclusion, Avoid race Condition and implement Synchronization between process

* Two operations Wait Signal

- * Semaphores are used to implement critical sections. By using Semaphores process can coordinate access to shared resource, such as shared memory or I/O device.
- * Semaphores are of two types
 - ▷ Binary Semaphores (0/1)
 - ▷ Counting Semaphores (n)

(Q) What is Swap in and Swap out?

Ans Swapping is a memory management scheme in which process can be temporary swap from main memory to secondary memory.

Swap In: Swap-in is a method of removing a program from a hard disk and putting it back into the main memory or RAM.

Swap-out: Swap-out is a process removing a program from a RAM and placing back in hard disk.

Q) Differential between Starvation and Aging.

Ans

Starvation: Starvation or indefinite blocking is a phenomenon associated with priority scheduling Algorithm in which a process ready for the CPU (resource) can wait to run indefinitely because of low priority.

Aging: Aging is a technique of gradually increasing the priority of process that wait in a system for long time.

Q) Explain Page Replacement Algoith LRU.

Ans The LRU Stands for the Least Recently Used. It keeps track of page usage in the memory over a short period of time. It works on the concept that page that have been highly used in the past are likely significantly used again in the future.

Q) What is inner process Communication , types and which one is fast and why ?

Ans Inter-process Communication is a mechanism provided by the OS for communication between several process. Its simply called IPC and can be control by some

Control mechanism and Communication Process.
IPC comes under Cooperating process because it can provide information sharing, computational speed, modularity and ease of data access.

Different types of Interprocess Communication

- * Shared Memory (very fast)
- * Message Passing
- * Message Queue
- * FIFO
- * Direct Communication
- * Indirect communication
 - * Synchronous Communication
- * Memory mapping
- * Sockets
- * Semaphores
- * file

multiple process
can communicate
with each other
in shared
memory

Q) What is Fragmentation? Define External fragmentation.

Ans fragmentation is basically a process in which free memory space is broken into little pieces. In this memory cannot be allocated to process due to small size and such block remain unused. It basically occurs in dynamic memory allocation system when many free block are too small to satisfy.

Internal fragmentation: When memory allocated to one process is somewhat larger than the request memory (dynamic Allocation can fix this)

External fragmentation: External fragmentation

happens when there's sufficient quantity of area within the memory to satisfy the memory request but this is represent in non-contiguous manner.

Q) Calculate the number of processes generated using N fork() statements?

Ans 2^N

Q) What is Realtime OS?

Ans Realtime Operating System (RTOS)
are used in environment where large number of events, mostly external to the Computer System, must be accepted and processed in short time or within certain deadline. Example Industrial Control, telephonic Switching or flight control etc. These Systems are time bounded.

Types of Realtime OS

- * Hard RTOS (Time is important)
- * Soft RTOS (Little relaxation in time)
- * Firm RTOS

Advantages

- * Maximum Consumption
- * Task shifting
- * focus on Application
- * RTOS is a embedded os
- * Error free
- * Memory Allocation

Disadvantages

- * Limited Task
- * Use Heavy System Resources
- * Complex Algorithm

Q) Difference between Preemptive and non-preemptive algorithm?

Ans * In preemptive Scheduling the CPU is allocated to the process for limited amount of time whereas in non-preemptive the CPU is allocated until process gets terminated.

- * In Preemptive the executing process get interrupted by higher priority process whereas in non-preemptive the executing process is not disturbed.
- * In Preemptive there is overhead of switching the process from ready state to running state vice versa maintaining ready queue whereas in preemptive there is no overhead maintenance.
- * In Preemptive if higher priority keeps coming back from low priority process has delay in execution whereas, in non-preemptive if the process with higher burst time executes the process with lower burst time goes through starvation.
- * Preemptive Scheduling has to maintain the integrity of shared data that's why it is cost annouative which is not the case of non-preemptive scheduling.

Q/ If the time slice is greater than the execution time or largest execution time process then round robin act as first come first serve (FCFS)

Readers-Writes problem.

Q) There is a file and 5 processes. How can you grant access so that only 2 processes can write to file and 1 can read file at a time

Ans To write a file we can use Counting Semaphores with count 2 when 2 two reader are in the critical section the counter is zero and other process are not allowed to write.

for reading we can use binary semaphores as only one reading process is allowed.

Q) Difference between Stack pointer and frame pointer.

Ans A Stack pointer defines the end of the current frame whereas a frame pointer defines the end of last frame.

Q) Explain Spooling.

Ans Spooling is a process in which data is temporarily held to be used and executed by a device program or system. Data is sent to and stored in memory or other volatile storage until the program or

Computer request it for execution.

SPOOL (Simulation peripheral Operation online)

- * Handles I/O device data Spooling as devices have different data access rate.
- * Maintaining the Spooling buffer, which provides a waiting station where data can wait while the slower device catches up.
- * Maintain parallel computation because of the Spooling process as a computer can perform I/O in parallel order.

③ Static and dynamic memory allocation

Ans : Stack/Static Memory Allocation

- * It's a temporary memory allocation scheme where data is accessible only if the method() that contains it is currently running.
- * It allocates and deallocates the memory automatically as soon as the corresponding method completes its execution.

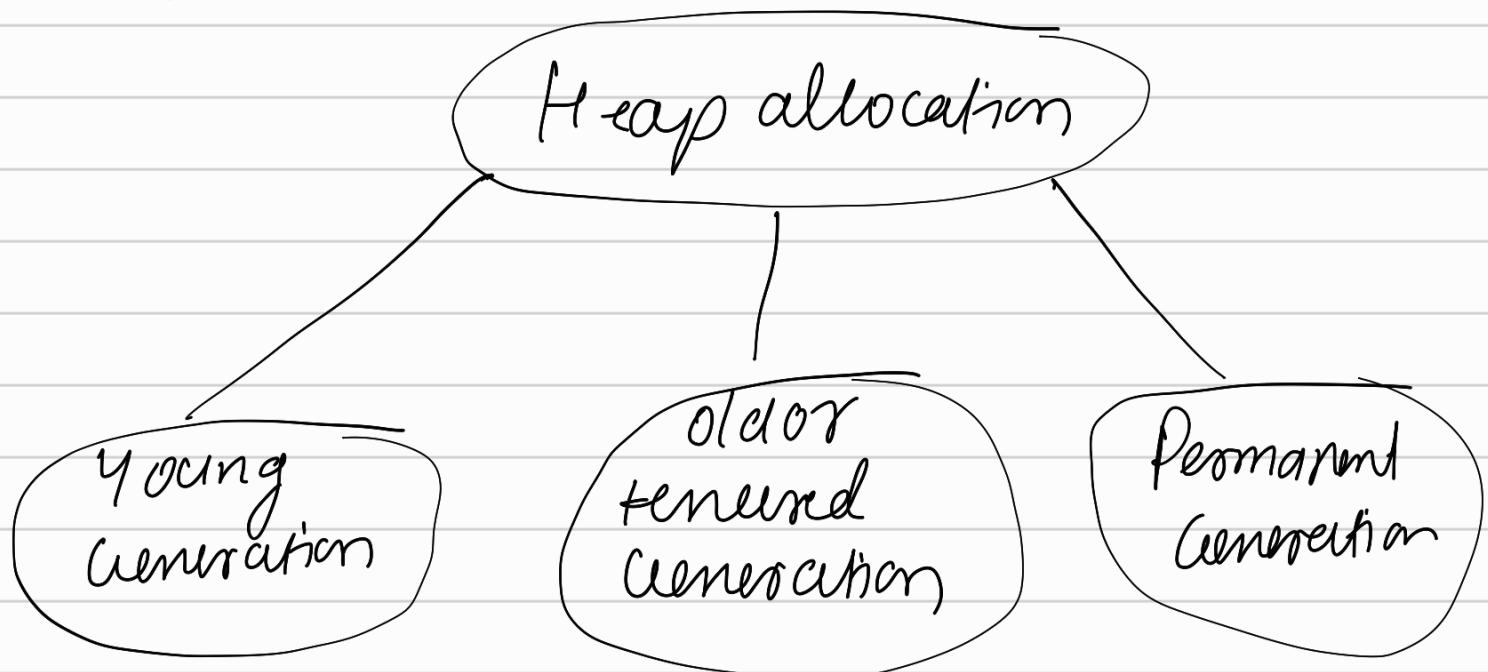
- * We will get error if stack memory is full.
- * Stack memory allocation is considered safer as compared to heap memory allocation as data stored can only be accessed by owner thread.
- * Memory allocation and deallocation are faster as compare to heap memory allocation.
- * It has less storage space as compare to heap memory.

■ Heap/Dynamic memory Allocation

- * This also throws error when memory is full.
- * This memory Space allocation Schema is different from stack memory Allocation here no automatic deallocation feature is provided. We need to use a Garbage collector to remove the old unused object in order to use memory efficiently.
- * The processing time (Accessing time) of this memory is quite slow as compared

To stack memory.

- * Heap memory is not threaded safe as stack memory because data stored in heap memory are visible to all threads.
- * The size of heap memory are quite large as compared to stack memory
- * Heap memory is accessible or exists as long as the whole application runs.



Q) Explain and give an example of a function pointer.

Ans * The function pointer is used to point functions, similarly the pointers are used to point variables.

- * It is utilized to save functions address.
- * The function pointer is either used to call the function or it can be sent as an argument to another function.

Syntax

return type (*functionptr) (parameters)

Referencing: When a pointer is allocated the address of the function to be associated with it then this process is referred to as referencing.

Dereferencing: When we use the (*) operator to get the value stored in the pointer.

* Code in Zepo with Example

Q) Difference between TCP and UDP headers.

Ans

TCP

- * Dynamic header (20-60) bytes.
- * It is connection oriented with the help of sequence numbers.
- * Flow control
- * Slow (seq no are waiting for all seq to come and send in sequence)
- * Checksum is mandatory
- * Error control
- * It does not support multicasting or broadcasting
- * Does not depend on ICMP protocol
- * HTTP, FTP, SMTP, telnet

UDP

- * Fixed header 8 bytes
- * It is connectionless no sequence number
- * Has no flow control (each datagram is independent)
- * fast (each datagram is independent & not wait for another datagram to come)
- * Checksum is optional
- * UDP has no error control
- * It supports multicasting and broadcasting
- * Depends on ICMP protocol
- * DNS, TFTP, SNMP

Q) What is inline function?

Ans C++ provides inline function to reduce the function call overhead. An inline function is a function that is expanded in line when it called. When a inline function is called whole code gets inserted or substituted at the point of inline Function Call. This Substitution is performed by the C++ compiler at compile time. An inline function may increase efficiency if its small.

Compiler may not Perform inlining in such circumstances:

- * If a function contains a loop (for, while and do-while)
- * If a function contain static Variable
- * If a function is recursive.
- * If a function return type is other than void and the return statement doesnot exist in the function body
- * If a function contain Switch or goto Statement.

Q) What is meant by early binding and late binding.

Ans

Binding

Binding refers to the process of converting identifiers (such as variables and procedure names) into addresses.

Binding is done for both variables and functions. For functions, it means that matching the call with right function definition by the Compiler. It takes place at runtime or compile time.

(Compile time)

Early Binding: A Compiler (or linker) directly associates an address to the function call. It replaces the call with a machine language instruction that tells the mainframe to leap to the address of the function.

Late Binding: (Runtime Polymorphism):

The Compiler adds code that identifies the kind of object at runtime then matches the call with the right function definition. This can be achieved by declaring Virtual function.

Q) State Various protocol in Various layers of TCP/IP.

Ans

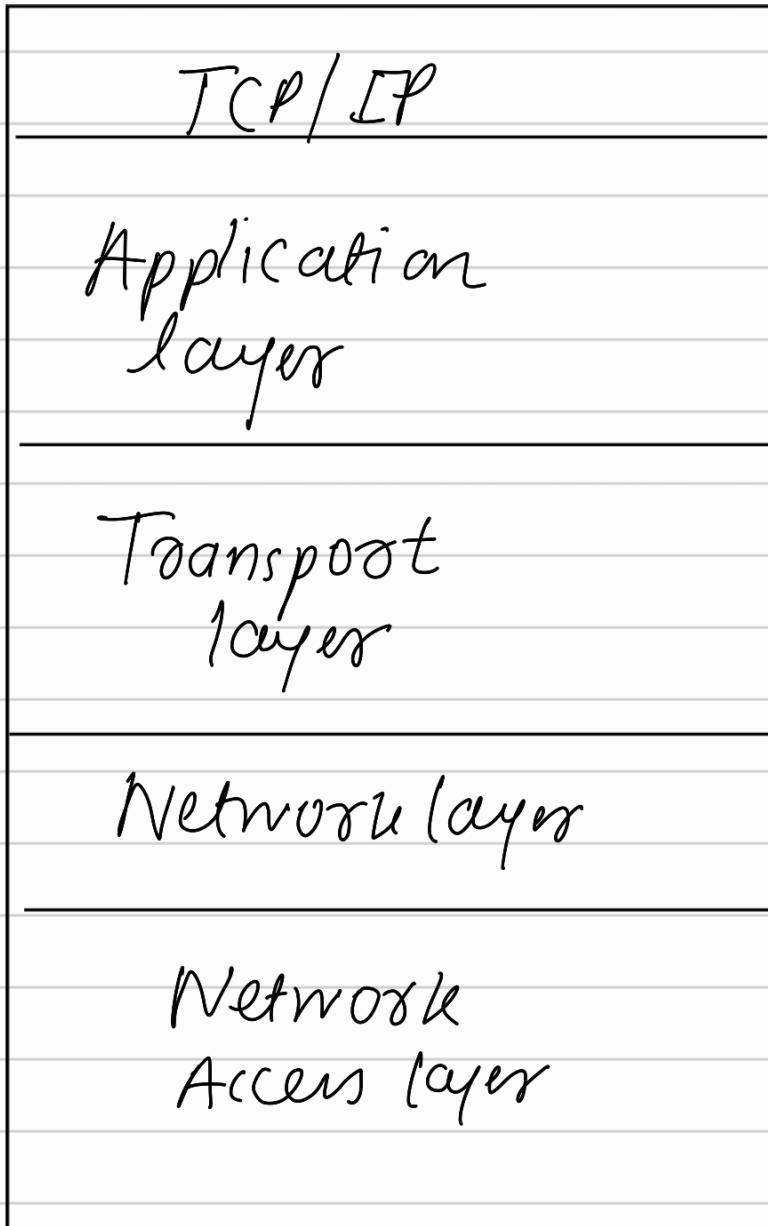
OSI

Application
presentation
Datalink
layer

Transport
layer

Network
layer

Datalink/
physical
layer



- Datalink layer

- * Packet's network protocol
- * Error prevention and framing
- * Point-to-point protocol (PPP).
- * framing and Ethernet IEEE 802.2 framing

are two examples of datalink layer protocols.

- Internet layer / Network layer

- * Defines the protocols which are responsible for the logical transmission of data over the entire network.
- * IP : Internet protocol is responsible for delivering the packets from source host to destination host by looking at IP address of destination in the packet head. Two versions of IP \Rightarrow IPv4 \Rightarrow IPv6.
- * ICMP : Internet Control message protocol. It encapsulates with IP datagram and is responsible for providing host with information about network problem.
- * ARP : Address Resolution Protocol. Its job is to find the address of a host from a known IP address. ARP has several types: Reverse ARP, Proxy ARP, Gratuitous ARP and Inverse ARP.

- Transport layer

* The TCP/EF transport layer protocols exchange data receipt acknowledgments and retransmits missing packets and ensure that packets arrive in order and without errors. (End-to-End communication)

* TCP: Applications can interact with

each other as they were physically connected by circuits.

* TCP transmits data in a way that resembles character-by-character transmission rather than separate packets.

* UDP: (Datagram delivery service).

* Connection between receiving and sending host is not verified by UDP.

* Application that transmit little amount of data use UDP rather than TCP because it eliminates the process of establishing and validating connections.

① Application Layer Protocols

* HTTP * SSH * NTP

NTP: (Network time protocol) it is used

to synchronize clocks on our computer to one standard time source. It is very useful like in situation like bank transactions.

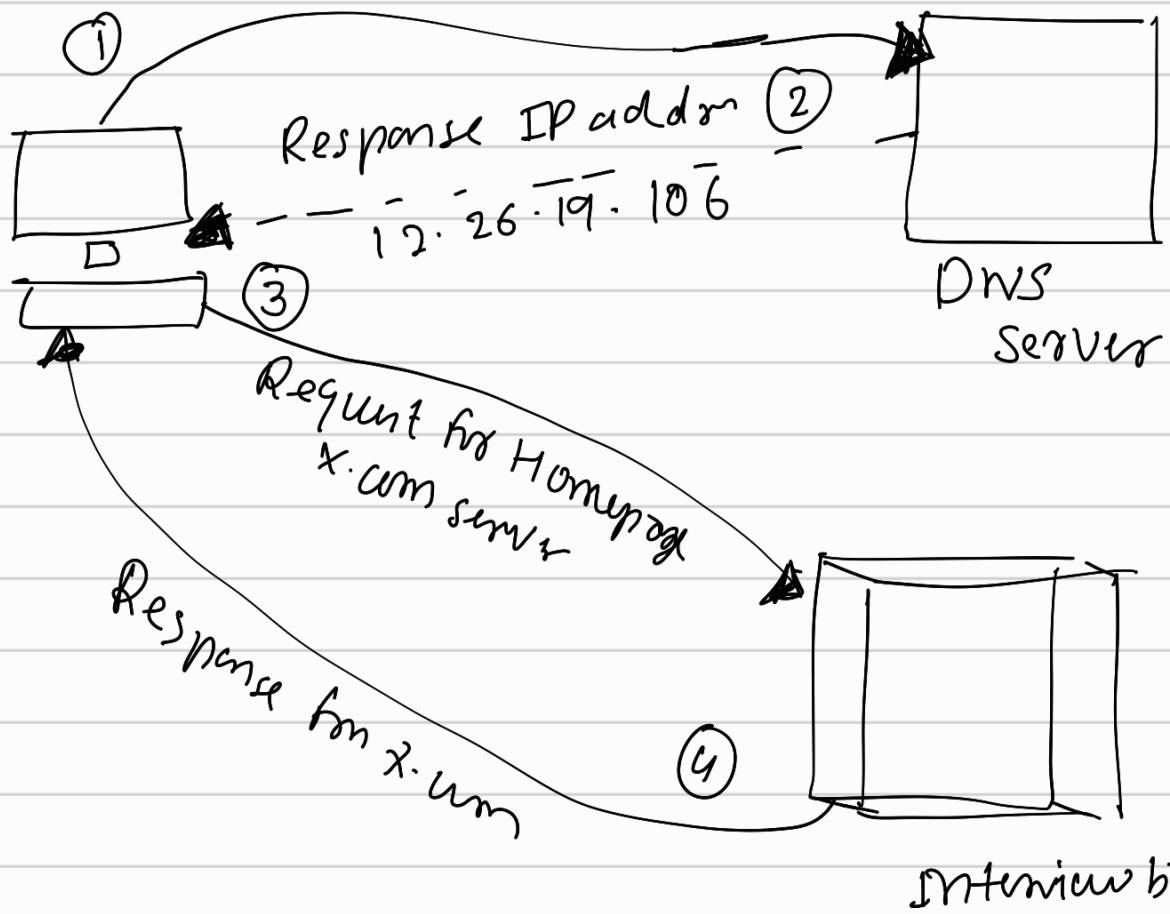
Q) Explain client server architecture.

Ans client: The client can be anything that request something from the server.

Server: Server is the computer that is designed to serve the request to the client.

A client server relationship responds to the request-response pattern and should adhere to a standard communication protocol that defines the language and rules for communication.
The Client Server Communication adheres to the TCP protocol suite. Client/Server messages are exchanged via TCP protocol until the connection is completed.

Request for IP address of x.com



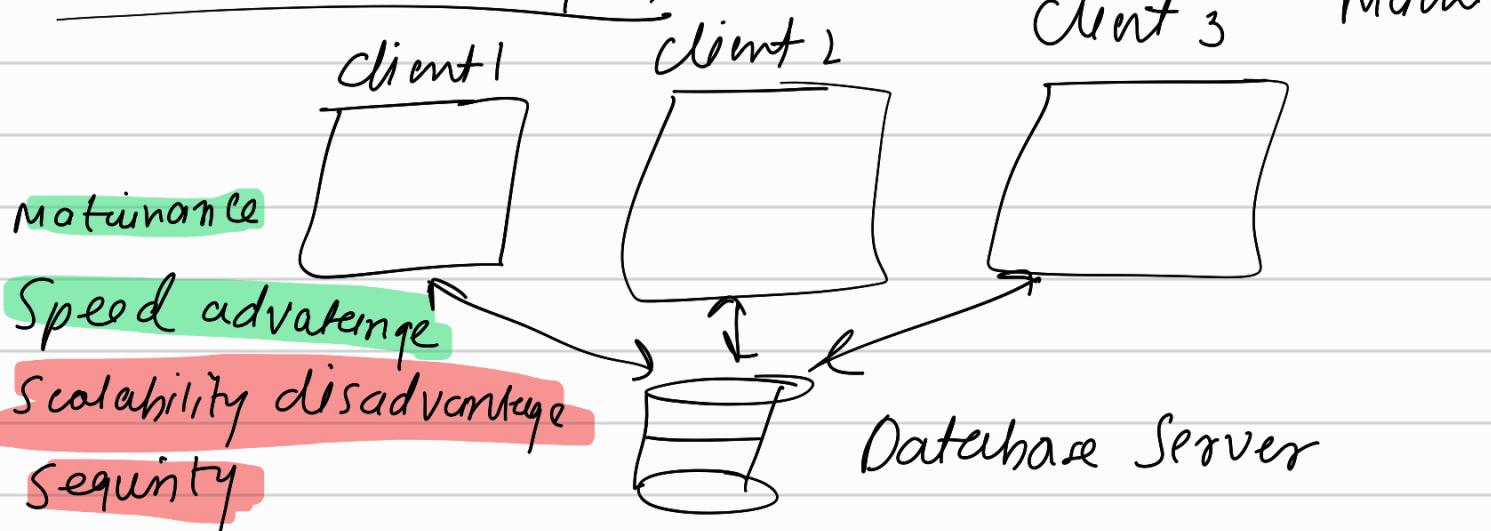
Types of Client-Server Architecture

1-tier Architecture

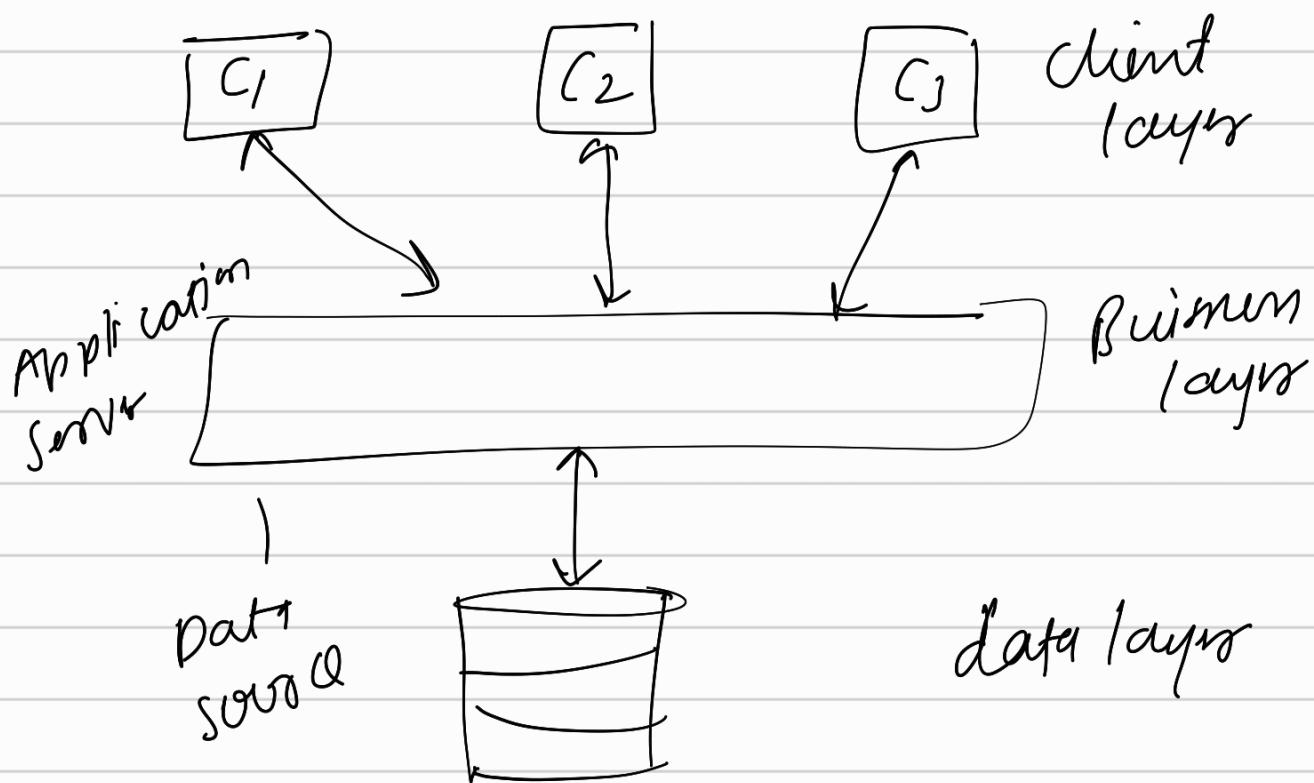
2-tier Architecture

3-tier Architecture

2-Tier (2 layers)



3 tier (3 layer)



Advantage **Security, Scalability**

Disadvantage **Maintainance**.

Q) What is L-Value and R-Value?

Ans

* A L-value / left-value / location-value is an assignable object. It is any expression that may occur on the left side of an assignment. Variables are obvious l-value but so are items in lists.

* A right-value/r-value is any expression that has a value that may appear on the right of an assignment

Q) Principles of OOPS

- 1) Abstraction
- 2) Encapsulation
- 3) Inheritance
- 4) Polymorphism.

Q) Define

- 1) MFC
- 2) COM
- 3) DCOM

* COM (Component object model)

* It is an interface standard designed for Software component. It helps for interprocess communication irrespective of the programming language to be used.

* COM is termed as a Software architecture that allows system to be built from

the different software vendors.

- * DCOM (Distributed Component object model)

Specially designed for distributed application. This supports the need in which components needs to communicate across network supported computer. Advantage of use is it have distributed garbage collecting that enhance CPU utilization.

- * MFC - Library / Framework used to build Win32 apps. Typically you will use it for building desktop applications.

Q) Explain VTABLE and VPTR?

Ans Vtable contains the address of virtual function. The compiler creates a Vtable for each class that contains virtual functions and for classes derived from it. The Vtable contains the addresses in order in which virtual functions are defined within the class. Whenever we create an object in class, the class gets loaded into memory and the Vtable gets created.

The address of the Value stored in the
Object Vptr.

- Q) full form COWIO → Console input/output
- Q) Explain Abstract and pure virtual function.

Ans Sometimes implementation of all functions cannot be provided in the base class because we don't know what would be implementation of the function in derived class. Such class is known as Abstract class.

In Abstract class we define abstract function for which we need to provide definition in derived class (imp step) such function which are denoted by virtual keyword is known as pure virtual function.
(Implementation in repo)

Ex Class phone {

Virtual void os();

}

Class Android :: public phone

{

void os()

{

cout << "This is iphone";

}

?

Class Apple :: public phone

{

void os()

{

cout << "This is Ios";

?

?

Q) Difference between const and static variables?

Ans Static Variables:

- Initialized only once

- Static Variables are for classes (not for object) i.e memory is allocated only once per class and every instance uses it so if one object modifies its value then the modified value is visible to other object as well.
- Value persists between different function call.

Const Variable

- Const Variables are a promise that you are not going to change its value anywhere in the program. If you it will complain.

Q) Define Scheduling. Which data structure is used in Scheduling.

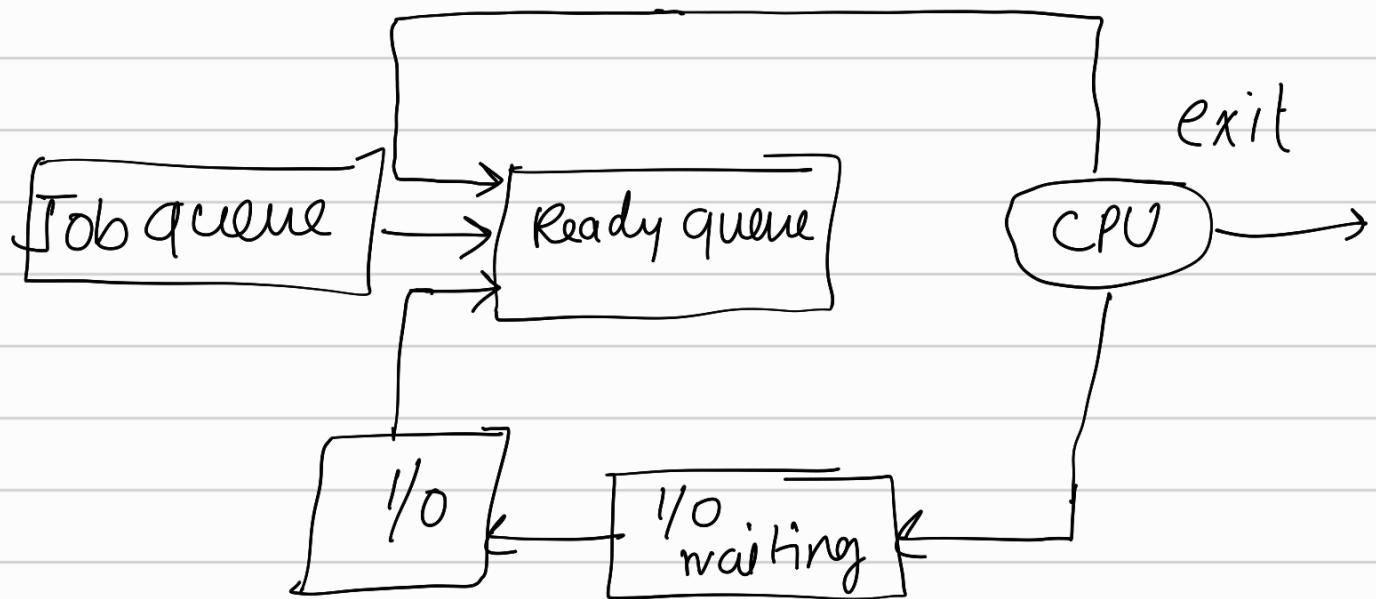
Ans The process Scheduling is the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process on the basis of a particular strategy. It is the essential part of multiprogramming operating system.

Category of Process Scheduling

Non preemptive: Process will execute completely and then removed.

Preemptive: Process will run for some time quantum and then replaced with other process after some brought back.

Process Scheduling Queue  ↪ data structure used here



Q) Where are the local, global, static, auto, register, extern, const, volatile variables stored?

Ans * **local** and **Auto** are stored in

Stack

- * Global and static Variables are stored in a DATA page/segments
- * Register Variables are stored in a register on the CPU if possible otherwise in stack.
- * Extern, Const & volatile are not specified they keep where compiler will tell them.

// There is no concept of virtual construct in C++

Q) What is virtual destructor? Explain the use of it in C++

Ans As we know, a destructor is implicitly involved when an object of a class goes out of scope or the object scope ends to free up the memory occupied by that objects.

Due to Early binding, when the object pointer of parent class is deleted, which was pointing to the object

In the derived class then, only the destructor of the parent class is involved if it does not involve the destructor of the child class, which leads to problem of memory leak in our program.

So when we use a virtual destructor, i.e. virtual keyword preceded by tilde(~) sign and destructor name, inside the parent class, it ensures that first child class's destructor should be invoked and then, the destructor of the parent class is called so that it releases the memory occupied by both destructors.

Q) What is a void pointer, a smart pointer, a wild pointer, a null ptr, and a dangling pointer? In what case are they used?

Ans Dangling Pointer: A dangling pointer

is a pointer that points to a deleted (or freed) memory location. Pointers can be dangling pointers in three ways:

- 1) De-allocation of memory
- 2) Function call
- 3) Variable goes out of scope

Void Pointer: Void pointer is a specific pointer type - void^* - a pointer that points to a unspecified data location in storage. The type is referred to as void.

Imp Consideration :

- * It is not possible to dereference void pointer. However it is possible to do so by typecasting on void pointer.
- * Due to lack of concrete value and thus size, pointer arithmetic is not possible on void pointer.

Null pointer: $\text{Int } * \text{ptr} = \text{NULL};$

Wild pointer: A wild pointer is one that has not been initialised to anything (not even NULL). The pointer may be set to a non-NULL garbage value that is not valid address.

`int *p; // wild pointer`

Smart Pointers: As we know we have memory leak problem in C++ so in smart pointer doesn't have to worry about any memory leak. C++ comes with its own mechanism that's smart pointer.