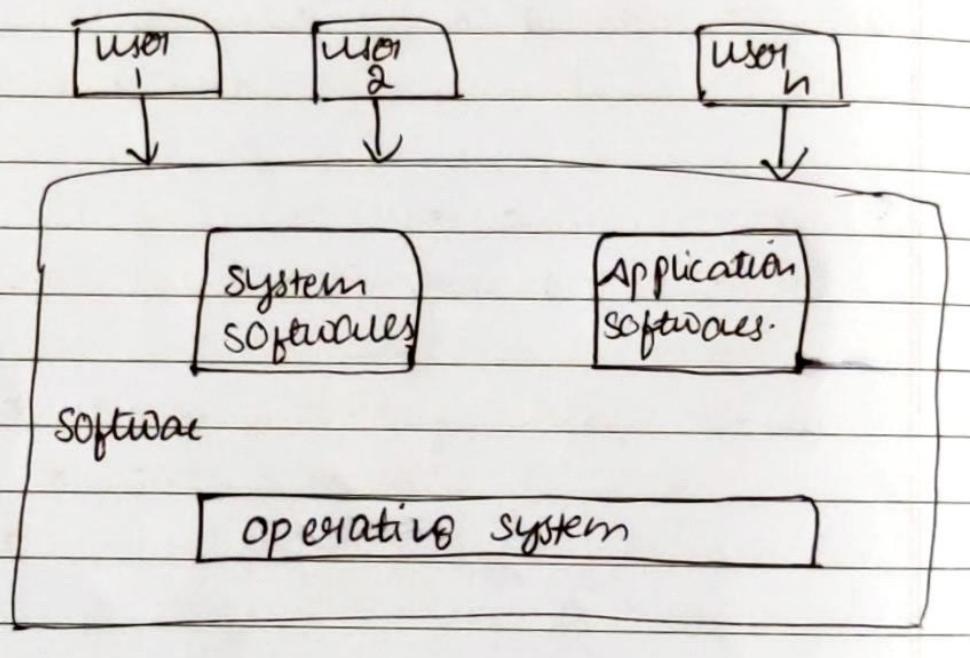


1. With neat diagram, explain the functions of the operating systems?

An operating sys is a program that acts as an interface b/w the user & computer hardware & controls the execution of all kinds of program.



- following are some important functions of O.S
- \* Memory management
  - \* processor management
  - \* device management
  - \* file management
  - \* security
  - \* control over S/m performance
  - \* job accounting
  - \* error detecting aids
  - \* coordination b/w other software, user

**Memory Management.**

It refers to management of primary memory

on main memory. Main memory is large array of words or bytes where each word or byte has its own address.

- \* keeps track of primary memory, i.e., what part of it are used by whom, what part are not in use.
- \* In multiprogramming, the OS decides which process will get memory when & how much.
- \* Allocates the memory when a process requests it to do so.

### processor management

In multi programming environment, the OS decides which process gets the processor when & for how much time this process calls<sup>d</sup> processes scheduling.

- \* keeps tracks of processor & status of process. The prog responsible for this task is known as traffic controller.
- \* allocates processor to a process
- \* de-allocates processor when process is no longer required.

### Device management

An OS manages device comm<sup>n</sup> via their respective drivers.

- \* keeps tracks of all devices. ~~it does~~ <sup>prog responsible</sup> for this task is known as I/O controller.
- \* decides which process gets the device when & for how much time.

### File management

A file sys is normally organised into directories for easy navigation & usage.

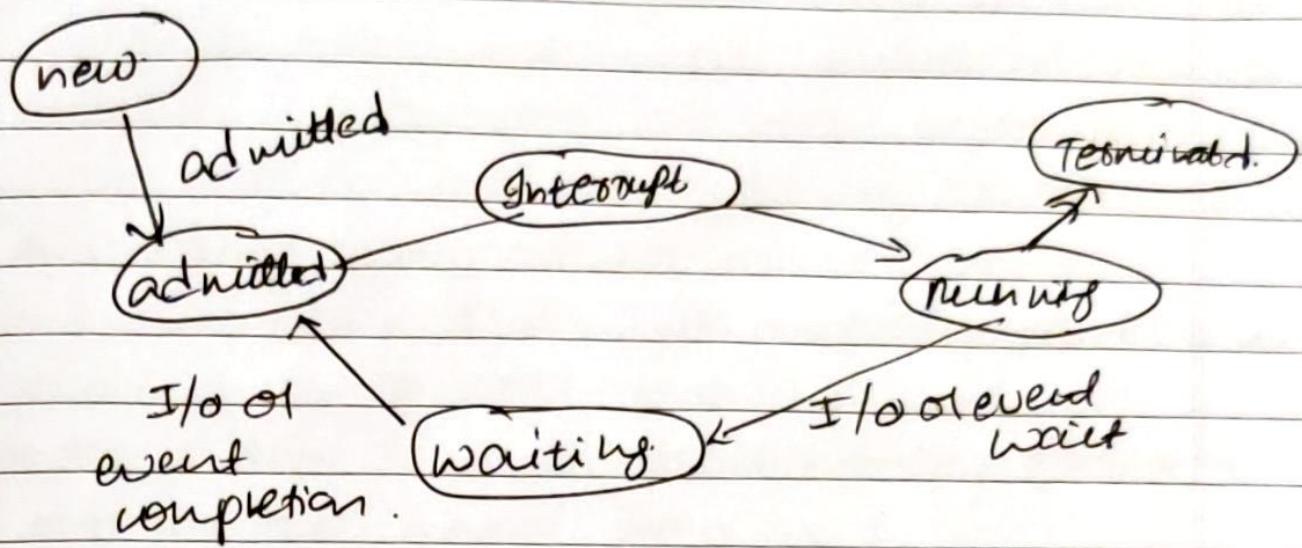
- \* decides who gets resources
- \* allocates the resources
- \* de-allocates the resources

- \* Security - By means of password & similar other techniques, it prevents unauthorised access to program & data.
  - \* control over S/m performance - recording delays b/w request for a service & response from the S/m.
  - \* job accounting - keeps track of time & resources used by various jobs & users.
  - \* error detecting aids - production of dumps, traces, errors messages & other debugging & error detecting aids.
- 2) list & explain briefly types of processing in operating systems.
- \* A process is basically a program in execution. The execution of a program must progress in sequential fashion.
  - \* A process is defined as an entity which represents basic unit of work to be implemented in the S/m.

### Different process state.

The following states of process are available.

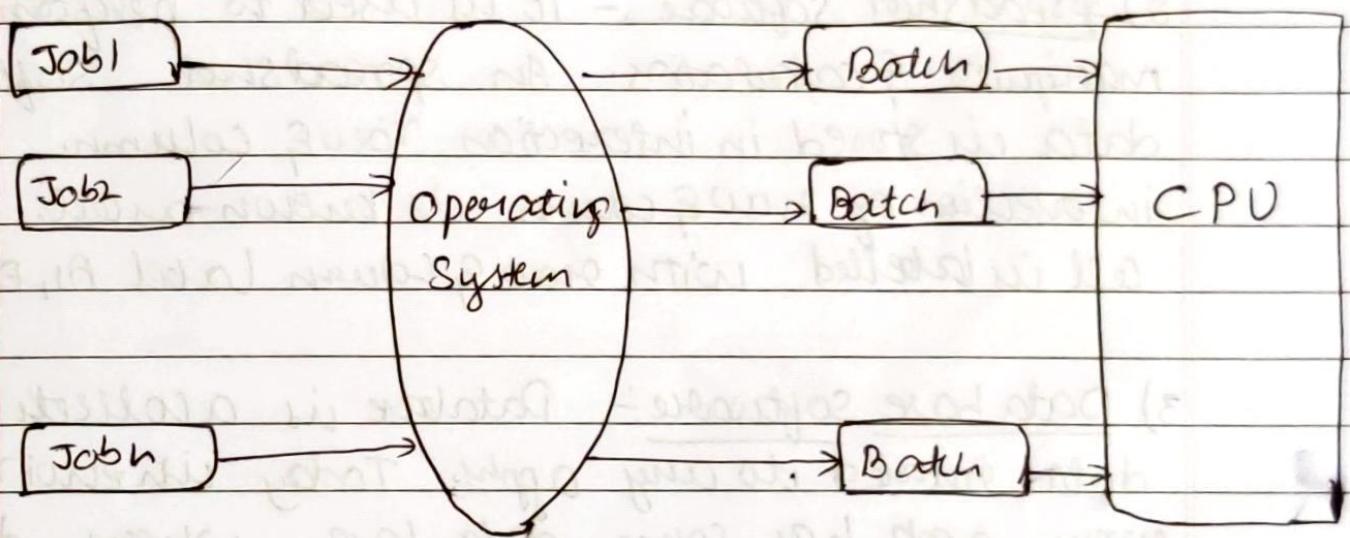
- i) New - The process is being created.
- ii) Ready - The process is waiting to be assigned to processor.
- iii) Running - Instructions are being executed.
- iv) Waiting - The process is waiting for some event to occur (such as an I/O completion or reception of a signal)
- v) Terminated - The process has finished executing.



3. With neat diagram, explain concept of batch processing.

Batch processing is a technique in which OS collects the programs & data together in a batch before processing starts. An OS does the following activities related to batch processing:

- \* The OS defines a job which has predefined seq of commands, pgm & data as a single unit.
- \* The OS keeps a no. of jobs in memory & executes them without any manual info.
- \* Jobs are processed in order of submission, i.e., first come first served fashion.



### Advantages.

- \* Batch processing takes much of work of operator to computer.
- \* Increased performance as a new job get started as soon as previous job is finished. without any manual intervention.

### Disadvantages.

- \* difficult to debug program.
- \* a job could enter an infinite loop
- \* due to lack of protection scheme, one batch job can affect pending jobs.

4. List & explain briefly about classification of Software applications

1) Presentation software - presentation software is a program to show info in form of slides. we can add text, graphics, video & images to slides to make them informative. the software has 3 components:

- \* text editor for inputting & formatting text.
- \* inserting graphics, text, video & other multimedia files.
- \* Slideshow to display the information.

2) Spreadsheet software - it is used to perform manipulations & calculations. In spreadsheet software data is stored in intersection row & column. The intersection of row & column is known as cell. The cell is labelled with row & column label A1, A2 etc.

3) Database software - Database is a collection of data related to any apps. Today in environment every app has some database where data regarding users stored. for this purpose, we use database software.

4) Multimedia Software:

it is a combination of text, graphics, audio & multimedia software used in editing of video, audio & text. Multimedia s/w used in growth of business, education, info, remote s/m & entertainment.

5) Simulation Software - simulation is a imitation of real world & environment. The simulation creates a physical environment of real world to represent similar behavior, func'n & key nature of selected topic. simulation is technology for education, engineering

testing, training & for scientific modelling of natural sm to gain insight into their functioning.

- word processing software - It is used to manipulate, format the txt, to create memos, letters, faxes & documents, processing sw is used to format & beautify the txt. It provides a list of features like thesaurus, the option provides synonyms, antonyms & related words chosen (or) phrase.

6) what are device drivers? Explain briefly its functions

- Device driver in computing refers to a special kind of software program of a specific type of software application which control a specific hardware devices for communication with the computer's operating system.

Device drivers are very essential for a computer S/m to work properly b/c without device driver the particular hardware fails to work according to its specification. In a very common way most term it as only driver also when someone says hardware driver that also refers to this device driver.

7) what is kernel? explain type of kernel briefly

The kernel is a computer program at the core of a computer's O.S with complete control over everything in the S/m.

The types of kernel as follows.

i) Monolithic Kernel:

In this type of kernel, all functions like process management, memory management, interrupt handling etc are performed in the kernel space.

To increase the performance of the S/m, a modular approach is made in the monolithic in which each function is present in a d/f module inside the kernel space.

ii) Microkernel:

In this type, the basic user services like device driver management, file S/m management etc are

present in user space.

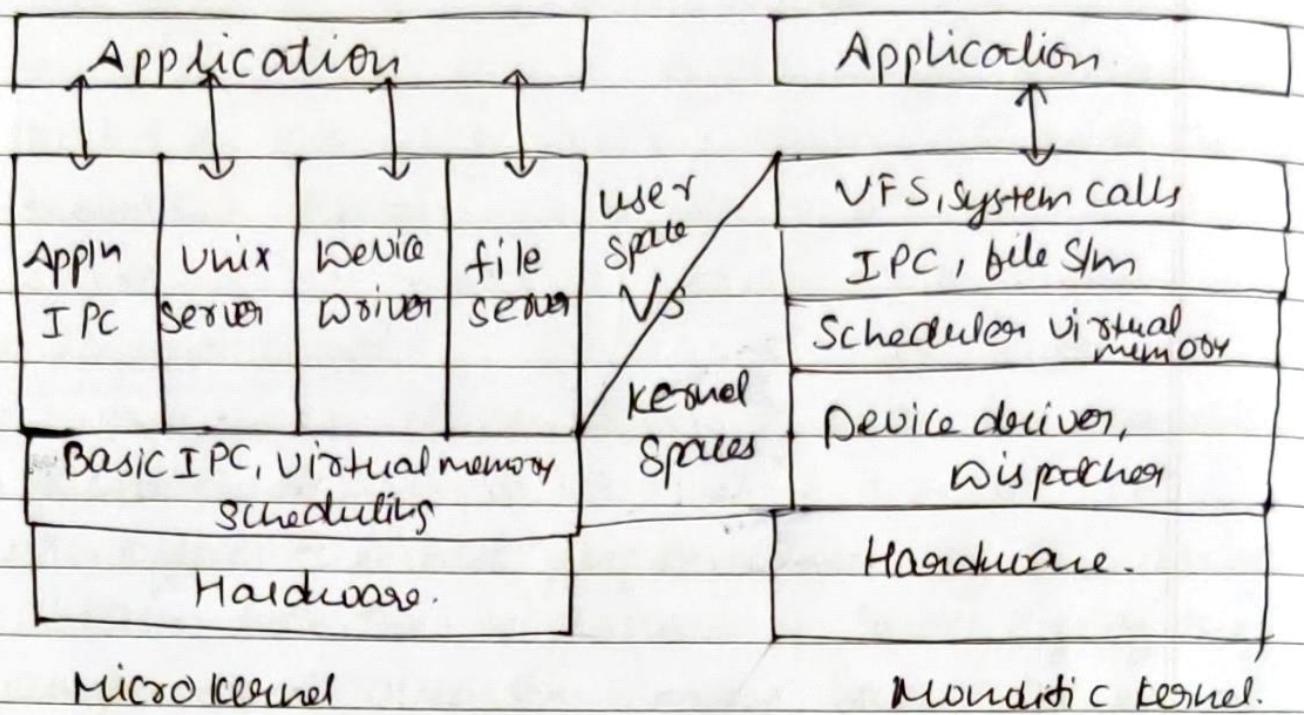
This type of kernel reduces the size of the kernel, but the speed of executing processes & providing other service is much slower than Monolithic kernels.

### ④ Hybrid Kernel:

This is a combination of both the Monolithic & Microkernel.

It offers the best performance of S/m, as we obtain both the high speed & small size of the kernel so that over S/m may have the maxm efficiency.

8. With neat diagram explain concept of monolithic vs Micro kernel.



Basis for comparison

1. Basic

Microkernel

In this user services & kernel services are kept in separate address space.

Monolithic kernel.

In this both user & kernel services are kept in the same address space.

2. Size

Microkernel are smaller in size

It is larger in size.

3. execution

Slow

Fast

4. extensible

easily extensible

Hard to extend

5. security

If a service crashes, it effect on working of microkernel

If a service crashes whole smcrashes in monolithic kernel

6. code

To write microkernel more code is required

To write a monolithic kernel less code is required

eg- QNX, Symbian & Linux      Linux, windows(95,98)  
K42, Mac OS X, minix      OS-9, AIX, HP-UX  
Coyotos, etc.      xts-400 etc.

a). what is virtualisation? how virtualisation helps?  
give the command to check the support of virtualisation by a s/m.

Virtualisation is a process of creating of shared or virtual representation of something, such as virtual apps, servers, storage, & w/w. It is single most effective way to reduce IT expenses while boosting efficiency & agility of all size business. ~~Virtualisation~~

Virtualisation can increase IT agility, flexibility & scalability while creating significant cost savings. Greater work load mobility, increased performance & availability of resources, automated operations- they're all benefits of virtualisation that make IT simpler to manage & less costly to own & operate.

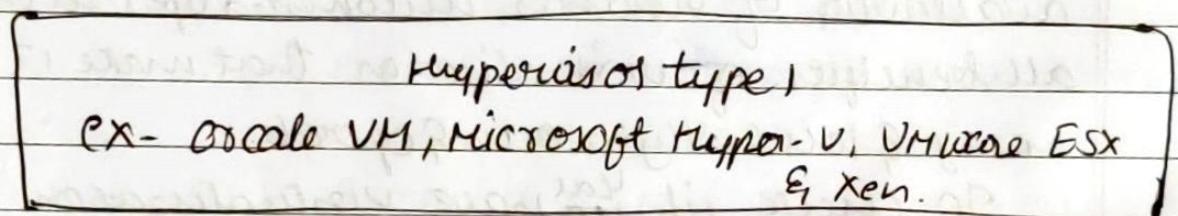
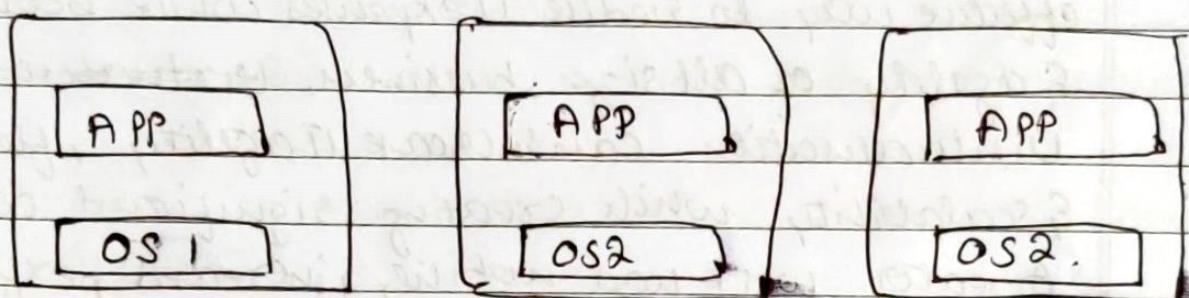
To check if ~~you~~ have virtualization technology support on your processor, open Task manager using **CTRL+SHIFT+ESC**. Now if your processor supports virtualisation, you will find it mentioned where other details are shown under performance tab. enabled or disabled & if your processor doesn't support virtualisation technology, you will not see virtualisation & Hyper-V mentioned anywhere at all.

10. with neat diagram, explain concept of Baremetal & hosted virtualisation

### Native on Bare Metal Hypervisor

Native hypervisors are software systems that run directly on the host's hardware to control the hardware & to monitor guest operating sys. The guest OS runs on separate level above the hypervisor. All of them have a virtual machine manager.

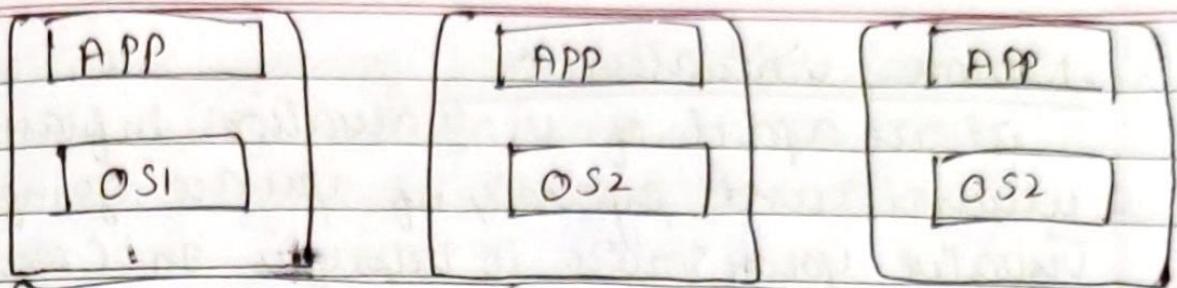
Ex of virtual machine architecture are Oracle VM, Microsoft Hyper-V, VMware ESX & Xen.



### Hosted hypervisor.

Hosted hypervisors are designed to run within a traditional OS. In other words, a hosted hypervisor adds a distinct software layer on top of host operating sys which guest OS becomes a 3rd SW level above hardware.

A well known ex of a hosted hypervisor is Oracle VM VirtualBox. Other include VMWare Server & workstation, Microsoft Virtual PC, KVM, QEMU & parallels.



Hypervisor 2

ex- VMware Server & Workstation, KVM, QEMU

OS installed Hardware

11. List the types of Virtualisation, Explain briefly.

- \* Server Virtualisation
- \* Client & Desktop virtualisation
- \* Services & apps Virtualisation
- \* Network Virtualisation
- \* Storage virtualisation.

### Server Virtualisation

It is virtualising your server infrastructure where you do not have to use any more physical servers for d/f purposes.

### Client & desktop virtualisation

This is similar to server virtualisation, but this time is on the user's site where you virtualise their desktops, we change their desktops with thin clients & by utilizing the datacenter resources.

### Services & applications Virtualisation

The virtualisation technology isolates apps from the underlying OS & from other apps in os, to increase compatibility & manageability. for ex - Docker can be used for that purpose.

## Network virtualisation

it is a part of virtualisation infrastructure, which is used especially if you are going to visualise your servers. It helps you in creating multiple switching, VLANs, NAT-ing etc.

## Storage Virtualisation

This is widely used in data centers where you have big storage to d/f hardware. This allocation is done through n/w connection. The leader on storage is SAN.

Q2) What are data types in C lang. write a C prog to check data types.

1) primary data types- These are fundamental data types in C namely integer (int), floating point (float) character (char) & void.

int - Integer : a whole no.

float - floating point value: i.e, a no with fractional part

double - a double precision floating point value.

char - a single character.

void - valueless special purpose type which we will examine closely in later sections.

2) Derived data types- they are nothing but primary data types but a little twisted or grouped together like array, structure, union & pointer.

→ array - an array is a collection of data items, all of the same type, accessed using a common name

→ structure - A structure in C ~~is~~ is collection of items of d/f types. You can think of structure as a "record" in Pascal or a class in Java without methods.

→ unions - A union is a special data type available in C that allows storing d/f data types

in same memory locations. You can define a union with many members, but only one member can contain a value at any given time.

pointer - A pointer is a variable whose value is the address of another variable, i.e., direct address of the memory location. like any variable or constant, you must declare a pointer before using it to store any variable address.

### C program:

```
#include < stdio.h>
#include < conio.h>
main()
```

{

closec()

```
printf ("short int is %2d bytes \n", sizeof (short int));
printf ("int is %2d bytes \n", sizeof (int));
printf ("int * is %2d bytes \n", sizeof (int *));
printf ("long int * is %2d bytes \n", sizeof (long int *));
printf ("unsigned int is %2d bytes \n", sizeof (unsigned int));
printf ("\n");

printf ("float is %2d bytes \n", sizeof (float));
printf ("float * is %2d bytes \n", sizeof (float *));
printf ("double is %2d bytes \n", sizeof (double));
printf ("long double is %2d bytes \n", sizeof (long double));
printf ("\n");

printf ("char is %2d bytes \n", sizeof (signed char));
printf ("char * is %2d bytes \n", sizeof (char *));
printf ("unsigned char is %2d bytes \n", sizeof (unsigned char));
getch();
```

by

13. what is padding byte. explain C how padding works.  
 In order to align the data in memory, one or more bytes are inserted (or left empty) b/w memory address which are allocated for other structure members while memory allocation. this concept is called structure padding.

Architecture of a computer processor is such a way that it can read 1 word from memory at a time. To make use of this advantage of processor.

struct student

{

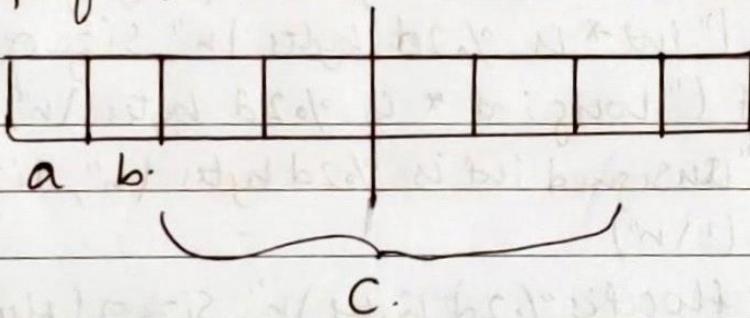
  char a;

  char b;

  int c;

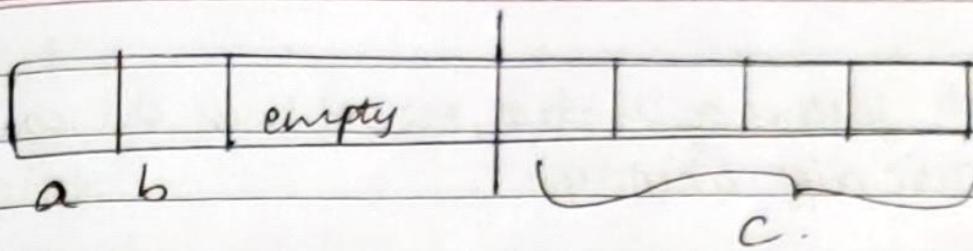
}

if we have 32 bit processor, then the pictorial representation of memory for above structure would be.



As we know that structure occupies the contiguous block of memory as shown in above diag. i.e., 1 byte for char a, 1 byte for char b, & 4 bytes for int c. Both char a & char b variables can be accessed in one CPU cycle, but we will face problem when we access int c variable as 2 CPU cycles are required to access value of 'c' variable.

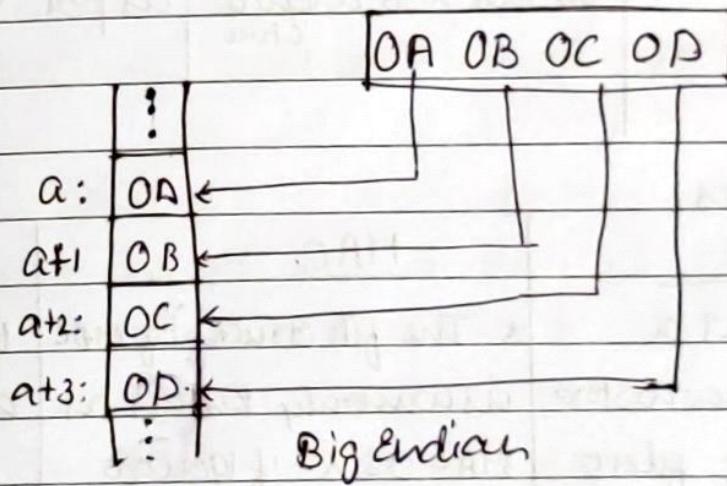
SO → In the first cycle 2 bytes & in the 2nd cycle other two bytes are accessed.



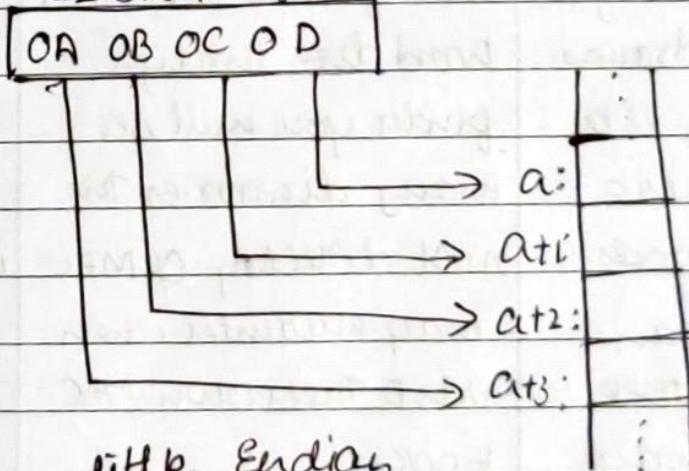
14. What is endianness? Explain its types with examples

Endianness refers to the order of storing & reading multiple bytes in memory. Endianness determines if the LSB of a word that we want to store in memory will go to lowest or highest address of assigned memory space.

There are two possibilities of Endianness: little Endian & big Endian.  
A big Endian stores the most significant byte of a word at smallest memory address & the LSB at largest.



32-bit integer



15. List features of Windows, MAC & Linux O.S, compare & contrast  
 \* Basic diff & history.

### Windows

\* It was released in 1985. it was supposed to be graphical user interface on top of MS DOS. all features of MS DOS later integrated in windows 95 release.

### MAC

This OS from apple stands older than windows. it was released in 1984. it began as graphical user interface right from its inception. in 2005 the design structure changed to intel x86 architecture.

### Linux

It was initially developed in university. it was released in 1991. designed for GNU developers. GNU developers later integrated onto Linux. it is open to everyone & every one can use as per specifications.

### \* File structures.

#### Windows

\* windows follows a directory structure to store the different kinds of files the user. It has logical drives & cabinet drives. it also has folders like documents, pics, music, videos & downloads. all these files can be stored in these folders & new folders can be created.

#### MAC

\* The file structure of MAC is commonly known as MAC OS X. if you go to dig into your MAC's hard disk through finder you will see many directories. The root directory of MAC may encounter when visit their own MAC book.

#### Linux

Linux has a completely different file structure from windows & MAC. it was developed with d/F code base. it stores data in form of tree. there is a single file tree & all your drives are mounted over this tree.

## \* Registry

### Windows

It is a master database which is used to store all settings on your computer. It is responsible to store all user info with passwords & device relate info. The registry also has an editor which allows you to view all keys & values or even drivers if necessary.

### MAC

MAC stores all appn settings in a series of config files which have various preferences folder in MAC. The plist file contains all properties in either plain text or binary format.

### Linux

It does not have a specific category of menu. All appn settings are stored on prog basis under the d/f users in the same hierarchy format of files being stored.

## \* Interchangeable Interfaces

### Windows

Windows interface was not interchangeable until windows 8. Windows Xp had some improvements but not Pro. Start menu, task bar, S/m tray & windows explorer.

### MAC

It has facility to bridge virtual network interfaces. This can be done by going to S/m preferences & managing the interfaces.

### Linux

It is easy to switch interfaces. You can switch the environment without having to carry all installations. There are utilities like GNOME & KDE which help in catering needs.

16. What is distributed O.S? Explain in detail.

Distributed O.S. is one of most important type of O.S. Multiple central processors are used by distributed S/m to serve multiple real time apps & multiple users. Accordingly data processing jobs are distributed among processors.

High speed buses.

A mechanism that transfers data b/w components inside a computer.

A DOS involves collection of autonomous computer sys which are able to communicate with each other through LAN/WAN.

This sys as mentioned above, incorporates various autonomous interconnected computers that communicate with each other using a shared communication n/w furthermore they are independent sys that possess their own memory unit & CPU.

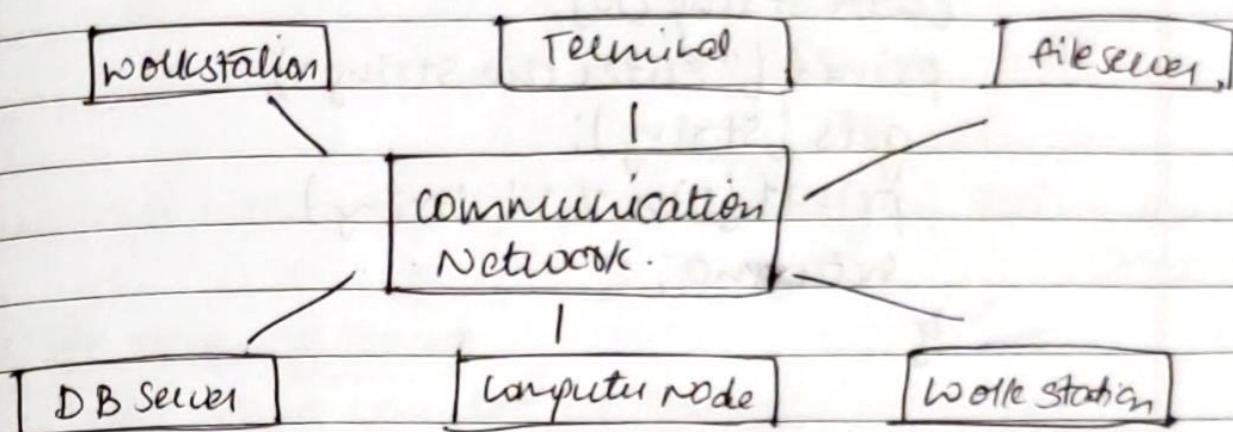
Another term which is used along side distributed O.S. is a loosely coupled sys. The processors in these S/m may differ in size & punch.

Distributed S/m can be considered to be more reliable than a central S/m b/c if S/m has only one instance of critical peripheral component like the CPU, n/w interface disk & so if that one instance fails, the S/m will go down completely.

Architecture of Distributed Operating Sys:-

- \* all S/w & hardware components are located remotely in order for them to communicate with each other.
- \* one of the most imp aspects of a distributed S/m is resource sharing. Resources are managed by servers & clients use these resources.

A DOS owns a no of independent sites which are connected through a communication n/w. However it is portrayed to user that they own with their own operating sys.



Q7. What is diff b/w gets vs scanf vs fgets. justify why  
 scanf is much inc:  
 gets is great while taking I/p such as  
 integer, character, float etc. it certainly falls  
 behind while taking strings I/p containing whitespace  
 Eg- #include <stdio.h>  
 int main()

```

    {
        char string[10];
        printf ("Enter String");
        scanf ("%s", string);
        printf ("%s", string);
        return 0;
    }
  
```

y.

gets() function in C

It is a pre-defined function in C which is used to read a string or a text line and store the I/p in a well-defined string variable. The function terminates its reading session as soon as it encounters a newline

character.

ex-

```
#include <stdio.h>
int main()
{
    char string[10];
    printf("Enter the string");
    gets(string);
    printf("\n %s", string);
    return 0;
}
```

fgets() function in C.

The standard C library also provides us with yet another function, the fgets() function. The function reads a text line or a string from the specified file or console. & then stores it into the respective string variable.

Similar to gets function, fgets also terminates reading whenever it encounters a newline character. But furthermore, unlike gets(), the function also stops, when EOF is reached or even if the string length exceeds limit, n-1.

Syntax. → fgets(char \*str, int n, FILE \*stream)

\* str - it is variable in which string is to be stored.

\* n - it is max ~~at~~ length of string should be read.

\* stream - it is file handle. from where the string to be read.

Ex-

```
#include <stdio.h>
int main()
{
    char string[20];
```

```

File *fp;
fp = fopen("file.txt", "r");
fscanf(string, 20, fp);
printf ("The string is: %s", string);
fclose(fp);
return 0;
}

```

B. with a neat ex. explain handling of strings in C language.

### Declaration of strings.

Here's how you declare strings:

```

char s[5];
s[0] s[1] s[2] s[3] s[4]
[ ] [ ] [ ] [ ] [ ]

```

Declaring string of 5 characters.

### initialisation of strings.

You can initialise string in no of ways.

```
char c[] = "abcd";
```

```
char c[50] = "abcd";
```

```
char c[] = { 'a', 'b', 'c', 'd', '\0' };
```

```
char c[5] = { 'a', 'b', 'c', 'd', '\0' };
```

### Assigning values to strings.

arrays & strings are second-class citizens in C; they don't support assignment operator once declared. for e.g.

```
char c[100];
```

```
c = "C programming";
```

### Read strings from user.

The `scanf()` func to read string.

The `scanf()` func reads the sequence of characters until it encounters whitespace (space, new line, tab).

ex-

scanf() to read a string

```
#include <stdio.h>
```

```
int main()
```

{

char name[20];

```
printf("enter the name")
```

```
scanf ("%s", name);
```

```
printf ("Your name is %s", name);
```

```
return 0;
```

}

- To read a line of text - we use fgets() function to read a line of string, & you can use puts() to display the string.

ex-2 fgets() & puts().

```
#include <stdio.h>
```

```
int main()
```

{

char name[30];

```
printf ("enter name");
```

```
fgets (name, sizeof(name), stdin);
```

```
printf ("Name: ");
```

```
puts(name);
```

```
return 0;
```

to print string we have to use puts(name);

passing strings to functions.

Strings can be passed to a function in a similar way as arrays

ex- passing string to a function.

```
#include <stdio.h>
```

```
void displayString (char str[]);
```

```
int main ()
```

```
{
```

```
char str[50];
```

```
printf ("Enter the string : ");
```

```
 fgets (str, size of (str), stdin);
```

```
displayString (str);
```

```
return 0;
```

```
}
```

```
void displayString (char str[])
```

```
{
```

```
printf ("String O/P : ");
```

```
puts (str);
```

```
}
```

Q. why the length of string is +1 for gets.

gets() reads in at most one less than size characters from stream & stores them into buffer pointed to by s. reading stops after an EOF or newline. A terminating null byte (\0) is stored after the last character in buffer. So it adds '\n' after your 4 letters, returning string - length + 1

ex-

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
int main (void)
```

```
{
```

```
char name [10] = {0};
```

```
printf ("Enter your name \n");
```

```
 fgets (name, 10, stdin);
```

```
printf ("Your name is %s & it is %d letters ", name,
```

```
strlen(name));
```

name [strcpy(name, "Ab")]=0;

printf("New- your name is %s. S%u id is %d letter is %c",  
name, strlen(name));  
letter=0;

y

### ~~strlen()~~ function

20 list string funcs available in C language

\* ~~strlen()~~ func<sup>n</sup> is used to find length <sup>live ex</sup> of character strings.

ex- int n;

char st[20] = "Bangalore";

\* n = strlen(st);

\* strcpy() func<sup>n</sup> copies contents of one string into another string.

Syntax- char \* strcpy(char \* dest<sup>n</sup>, const char \* source);

ex - strcpy(st1, st2) - Copies contents from st2 to st1

\* strcat() func<sup>n</sup> - in C language concatenates two given strings. it concatenates some strings at end of dest<sup>n</sup> string

Syntax- char \* strcat (char \* dest<sup>n</sup>, const char \* source);

\* strncat() - in C language concatenates portion of one string at end of another string.

Syntax- char \* strncat (char \* dest, const char \* source, size\_t num);

ex

strncat(st2, st1, 3); - first 3 characters of st1 is concatenated at end of st2.

### \* `strcmp()` -

It compares two given strings & returns zero if they are same. If length of string 1 < string 2, it returns < 0 value. If length of string 1 > string 2, it ret zero value.

Syntax - `int strcmp (const char* str1, const char* str2);`

### \* `strlwr()` - converts given string to lower case.

Syntax - `char * strlwr (char * string);`

`strlwr()` function is non stand function which may not available in Standard Library in C.

### \* `strupr()` function -

It converts a given string to uppercase.

Syntax - `char *strupr (char *string);`

### \* `strrev()`

`strrev()` function reverses a given string.

Syntax - `char *strrev (char *string);`

Ex -

`char name[20] = "fkl";`

then `strrev(name) = lkf`

### \* `& atoi()` function

It converts string value to numeric value & converts a numeric-string value to equivalent integer value.

Syntax - `int atoi(string);`

Ex -

`printf ("Output = %.d", atoi ("123") + atoi ("2347"));`

This printf() will print 357

### \* `ltoa()` function

Converts a long int string value to equivalent long.

integer value.

Syntax - long int l\_toll (string),

ex -

```
printf ("output = %.d", l_toll ("486384") -  
atol ("112233"));
```

This statement will print 374151

\* atof () func:

converts a floating point text format values to double value.

Syntax - int atoi (string)

```
ex - printf ("%f", atof ("3.1412") * 5 * 5);
```

\* strchr() function

strchr() function return pointer to first occurrence of the character in given string.

Syntax - char \* strchr (const char \* str, int character);