Q. Discuss on the OS Linux, windows and Mac OS.

→ Basic Difference and history -

→ windows - It was 1st released in 1985. It was supposed to be graphical user interface on top of MS DOS. All features of MS DOS were later integrated in windows 95 release. It was huge success in and lead to windows transition.

→ MAC - This OS from Apple stands older than windows. It was first released in 1984. It began as a graphical user interface right from its inception in 2005 the design & structure of MAC OS was changed to intel x86 based architecture

→ Linux - It was initially developed in Finnish university. It was released in 1991 and designed for GNU developers. GNU developers later integrated it into Linux. It is open to customers and everyone can use as per the specification

→ File structure -

→ windows - windows follows a directory structure to store different kinds of file of the user. And it has logical drives and cabinet drawers. It also has folders. Some common folders like documents, picture, music, video and downloads. All these files can be stored in these folders and also new folders can be created. It also has files which can be a spreadsheet (or) any application program. It can have extensions as .txt, .jpg etc. In addition to this windows also provides recycle bin where all deleted files can be stored. Recycle bin can be configured to increase its size.

→ MAC - This 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 they first time own MAC book.

→ Linux - Linux has a completely different file structure from windows and MAC. It was developed with different code base. It store data in the form of tree. There is a single file tree and all your drives are mounted over this tree.

→ Registry -

→ windows - windows registry is a master database which is used to store all settings on your computer. It is responsible to store all user information with its password and device relate information. the registry also has an editor which allows you to view all keys and values or even drivers if necessary

→ MAC - MAC stores all application settings in a series of .plist files which make the userrows preferences folder in MAC. this .plist file contains all properties in either plain text or binary format. these are stored at :/Library/ preferences folders.

→ Linux - Linux also does not have a specific registry of its own. All application settings are stored on program basis under the different users in the same hierarchy format of the files being stored. There is not centeralized database for storing these details and so periodic cleaning is also not required

→ Interchangeable Interface -

→ windows - windows interface was not interchangeable until windows 8. windows xp had some improvements but not par. start menu, task bar, System Tray and windows explorer

→ MAC - MAC has facility to bridge virtual n/w interface. this can be done by going to system perferences and managing the interfaces

→ Linux - Linux is easy to switch interface. you can switch the environment without having to carry all installations. There are utilities like GNOME and KDE which help in catering these needs. They help in focusing on different aspects

→ Command Terminal -

→ windows - Terminal or command prompt is a black box ideally used to execute commands. It is also called windows command processor. It is used to execute commands and different batch files. It can also be used for administrative functions and Trouble shoot and solve all windows issues

→ mac - mac provides a console as a terminal application. it has console command line, prompt and terminal. command line is used to type your commands. prompt will provide you some information and also enable you to run commands. terminal is the actual interface which will provide the modern graphical user interface as well.

→ linux - linux also provides a terminal - you can find terminal at - Applications -> System tools Applications -> utilities. in addition to this there is also shell prompt. the most common shell used in bash. it defines how the terminal will behave and look when it is run.

② Booting process

the following are the 6 high level stages of a typical linux boot process

BIOS	Basic input/output system executes MBR
MBR	Master Boot Record executes GRUB
GRUB	Grand unified Bootloader executes kernal
kernal	kernal executes /sbin/init
init	init executes runlevel programs
Runlevel	Runlevel programs are executed from /etc/rc.d/rc*.d/

→ BIOS - BIOS stands for Basic input/output system
- performs some system integrity checks
- Searches, loads, and executes the boot loader program
- It looks for boot loader in floppy, cd-rom or hard drive, you can press a key during the BIOS Standup to change the boot sequence
- once the boot loader program is detected and loaded into the memory, BIOS gives the control to it
- So, in simple terms BIOS loads and executes the MBR boot loader

→ MBR - MBR stands for master Boot process
- It is located in the 1st sector of the bootable drive.

- typically /dev/hda, (or) /dev/sda
- MBR is less than 512 bytes in size. This has 3 components
  (1) primary boot loader info in 1st 446 bytes
  (2) partition table info in next 64 bytes
  (3) mbr validation check in last 2 bytes
- It contains information about GRUB (or LILO in old systm)
- so, in simple terms MBR loads and executes the GRUB boot loader

(3) GRUB -
- GRUB stands for Grand unified Bootloader
- If you have multiple kernal images installed on your system, you can choose which one to be executed.
- GRUB displays a splash screen, waits for few seconds, if you don't enter anything, it loads the default kernel image as specified in the GRUB configuration file
- GRUB has the knowledge of the filesystem
- GRUB configuration file is /boot/grub.conf
- GRUB just loads and executes kernel and initrd images

(4) Kernel -
- mount the root file system as specified in the "root=" in grub.conf
- kernel executes the /sbin/init program
- Since init was the 1st program to be executed by Linux kernel, it has the process id (PID) of 1. Do ps -ef | grep init' and check the pid
- initrd stands for initial RAM Disk
- initrd is used by kernal as temporary root file system until kernal is booted and the real root file system is mounted. It also contains necessary drivers compiled inside, which helps it to access the hard drive partition, and other hardware

(5) init -
- looks at the /etc/inittab file to decide the Linux run level
- following are the available run levels
  - 0 - halt
  - 1 - Single user mode
  - 2 - Multiuser, without NFS
  - 3 - Full multiuser mode

- 4 - unused
- 5 - X11
- 6 - reboot

- init identifies the default run level from /etc/inittab and uses that to load all appropriate programs
- execute egrep initdefault /etc/inittab on your system to identify the default run level
- If you want to get into trouble, you can set the default run level to 0 or 6. Since you know what 0 and 6 means, probably you might not do that.
- Typically you would set the default run level to either 3 or 5

(6) Runlevel programs -

- When the Linux system is booting up, you might see various services getting started. For ex, it might say "starting sendmail --OK". Those are the runlevel programs, executed from the run level directory as defined by your run level.
- Depending on your default init level setting, system will execute the programs from one of the following directories.
  - Run level 0 - /etc/rc.d/rc0.d/
  - Run level 1 - /etc/rc.d/rc1.d/
  - Run level 2 - /etc/rc.d/rc2.d/
  - Run level 3 - /etc/rc.d/rc3.d/
  - Run level 4 - /etc/rc.d/rc4.d/
  - Run level 5 - /etc/rc.d/rc5.d/
  - Run level 6 - /etc/rc.d/rc.6.d/
- Please note that there are also symbolic links available for those directory under /etc directly. So, /etc/rc0.d is linked to /etc.rc.d/rc0.d.
- Under the /etc/rc.d/rc*.d/ directories, you would see programs that start with S and K
- Programs starts with S are used during startup. S for Startup.
- Programs starts with K are used during shutdown. K for kill

Functions of operating system

→ security - the operating system uses password protection to protect user data and similar other techniques. It also prevents unauthorised access to programs and user data.

→ control over system performance - monitors overall system health to help improve performance. records the response time between service requests and system response to give a complete view of the system health. This can help improve performance by providing important information needed to troubleshoot problems

→ job accounting - o s keeps track of time and resources used by various tasks and users, this info can be used to track resource usage for a particular user or group of users

→ error detecting aids - the o s constantly monitors the system to detect errors and avoid the malfunctioning of computer system

→ co-ordination btw other softwares and users - o.s also co-ordinates and assigns interpreters, compilers, assemblers, and other software to the various users of the computer system

→ memory management - the o s manages the primary memory or main memory. main memory is made up of a large array of bytes or words where each byte or word is assigned a certain address. main memory is fast storage and it can be accessed directly by the cpu. For a program to be executed, it should be first loaded in the main memory. ~~An o.s~~ ~~perform~~

An o.s performs the following activities for memory management

It keeps track of primary memory i.e, which bytes of memory are used by which user program. the memory address that have already been allocated and the memory address of the memory that has not yet been used.

It allocate the memory to a process when the process requests it and deallocates the memory when the process has terminated or is performing I/O operation

→ process management -

In a multi programming environment, the o s decides the order in which process have access to the processor, and how much processing time each process has this function of os is call process scheduling.

An o.s performs the following activities for processor management

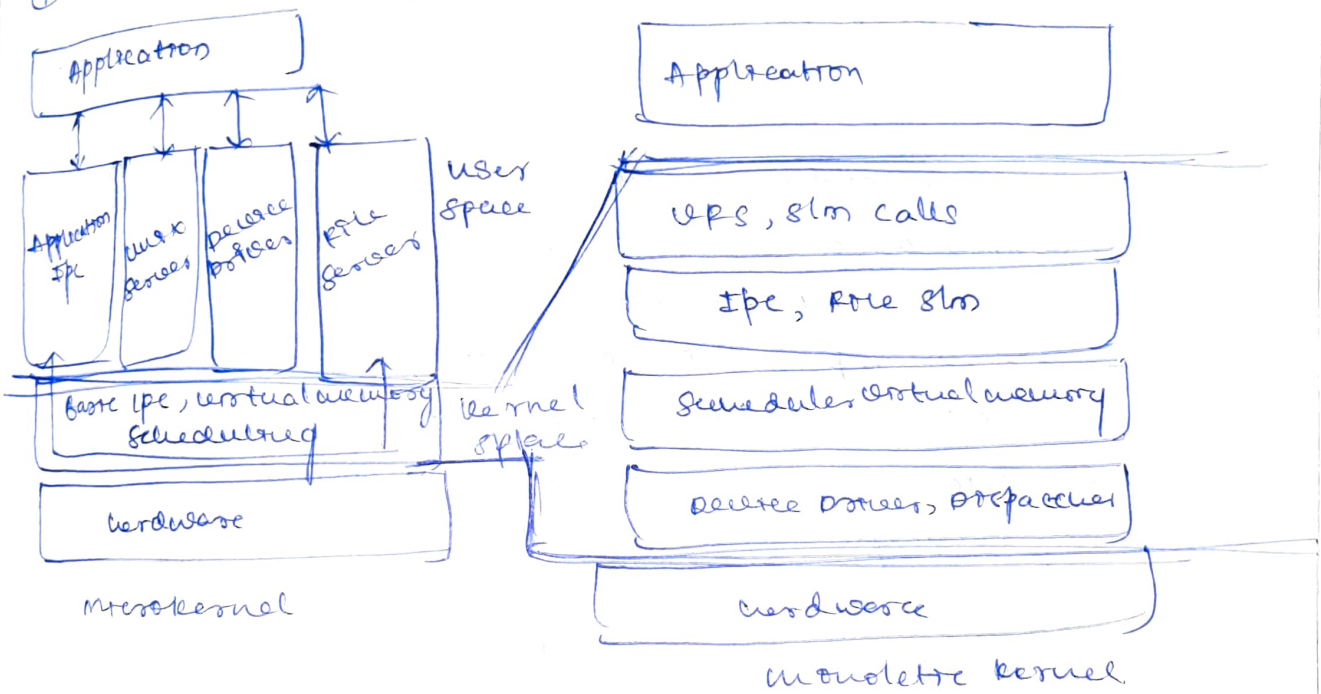
keeps track of the status of processes. the program which performs this task is known as a traffic controller. Allocates the cpu that is processor to a process. De allocates processor when a process is no more required

→ device management :-

An OS manages device communication via its respective drivers. It performs the following activities for device management.
Keeps track of all devices connected to the sys. designates a program responsible for every device known as the I/O controller.
Decides which process gets access to a certain device and for how long, allocates device in an effective and efficient way. Deallocates devices when they are no longer required.

→ File management :- A file system is organized into directories for efficient (or easy navigation and usage. These directories may contain other directories and other files. An O.S carries out the following file management activities. It keeps track of where information is stored, user access settings & status of every file, and more. These facilities are collectively known as the file sys.

④ Draw a neat diagram explain monolithic vs micro kernel.



→ kernel is the core part of the operating sys
→ It manages the sys resource
→ kernel is like a bridge between application and hardware of the computer.
→ The kernel can be classified further into 2 categories
→ microkernel and monolithic kernel.
→ Microkernel is the one in which user services and kernel services are kept in seperate address space.
→ However in monolithic kernel user services and kernel services both are kept in same address space.

	microkernel	monolithic kernel
Basic	- in microkernel user services and kernel, services are kept in seperate address space	- in monolithic kernels both user service and kernel services are kept in the same address space
Size	- microkernel are smaller in size	- monolithic kernel is larger than micro kernel
execution	- slow execution	- fast execution
extendible	- the microkernel is easily extendible	- the monolithic kernel is hard to extend.
Security	- if a service crashes, it does effect on working of microkernel	- if a service crashes the whole system crashes in monolithic kernel
code	- to write a microkernel, more code is required	- to write a monolithic kernel, less code is required.

(5) List the commands to check disk partition in windows

→ open a command prompt –
Start > Run > cmd

→ start up the DISKPART utility –
C:\users\Administrator> diskpart

→ Select the disk you want to view
(example uses disk 1, it can be any valid disk number)

DISKPART > Select disk 1

→ View the details of the Selected disk
DISKPART > details disk

(6) List the steps to check the disk partition to stop or Start the services on windows operating system

Unit-II Process Scheduling IPC

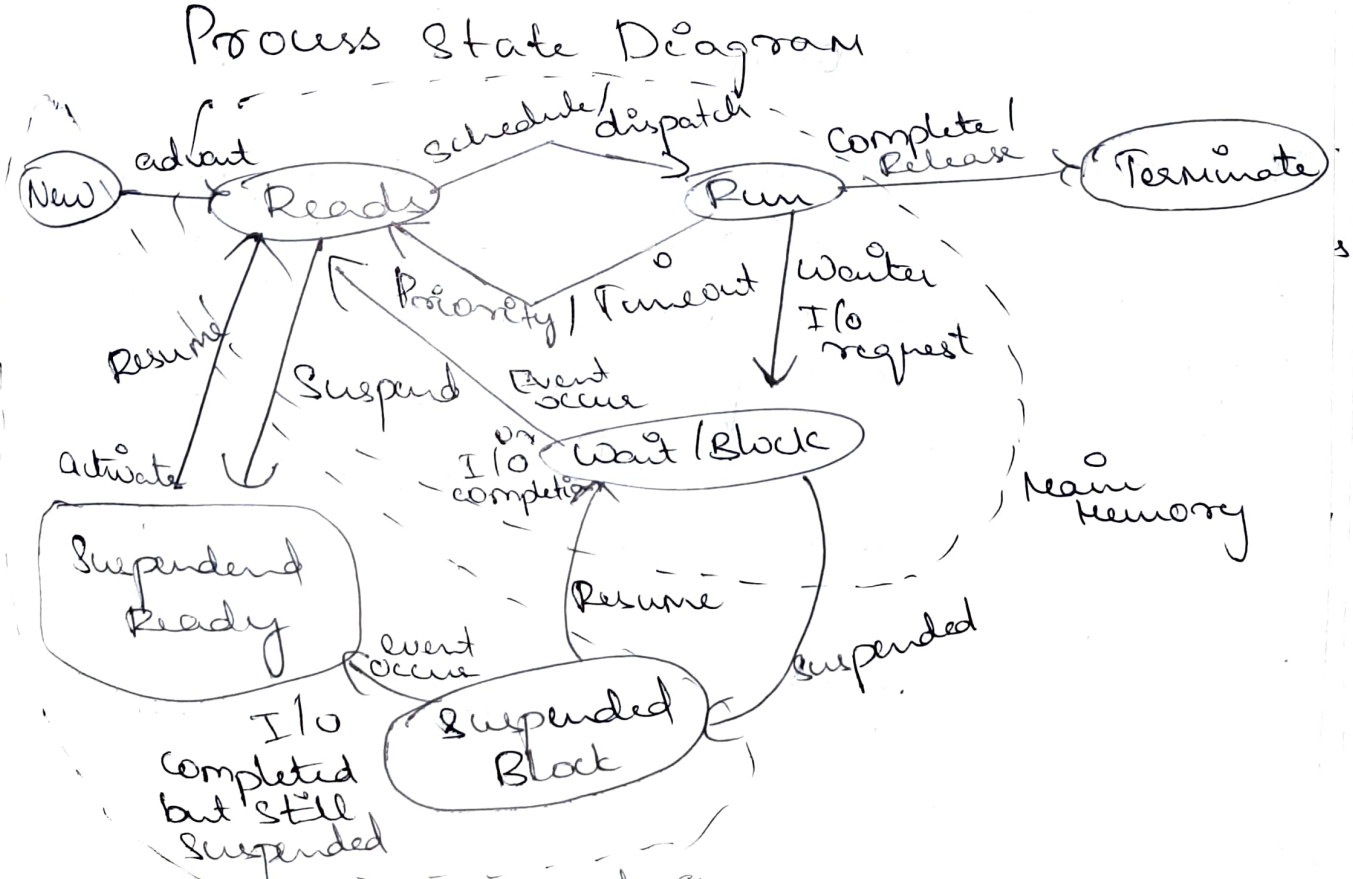
Process: A process is program (under) in execution
⇒ when we write a C program the compiler creates binary code, the original program and binary code both are programmed. when we actually run the binary code it becomes a process.

A process is an active entity program is a passive entity.

Attributes or Characteristics of a process

(i) Process Id
(ii) Process - State
(iii) CPU registers
(iv) I/O status information
(v) CPU scheduling info. etc

Process State Diagram



Process States:

- New (Create):- The process is about to created but not yet created, it is the program which is present in secondary memory that will picked by O.S to create the process.

- Ready :- New → Ready.

Ready to run: After the creation of process, the process enters the ready state i.e the process is loaded in to the main memory. The process is ready to run and is waiting to get the CPU time for its execution. These process are waiting in a queue is called Ready Queue.

Run :- The process is choosen by CPU for execution and the instructions within the process are executed by any one of the available CPU Cores.

Blocked or Wait :- Whenever the process requests access to I/O or needs input from the user or needs access to a critical region (the lock for which is already acquired) it enters the blocked or wait state. The process continues to wait in the main memory and doesnot require CPU. Once the I/O operation is completed the process goes to ready state.

Terminated or completed: Process execution completed, or process is killed as well as PCB is deleted

Suspend ready: The process that was initially in the ready state but were swapped out of main memory, and placed on to secondary memory by scheduler are said to be in suspended ready state.

The process will transition back to ready state whenever the process again brought onto the main memory.

Suspended wait or suspend blocked: Similar to suspend ready but user the process which was performing I/O operations and lack of main memory caused them to move to secondary memory.

When work is finished they may go to suspend ready. CPU and IO Bound Process.

If the process is intensive in terms of CPU operations then it is called CPU Bound Process.

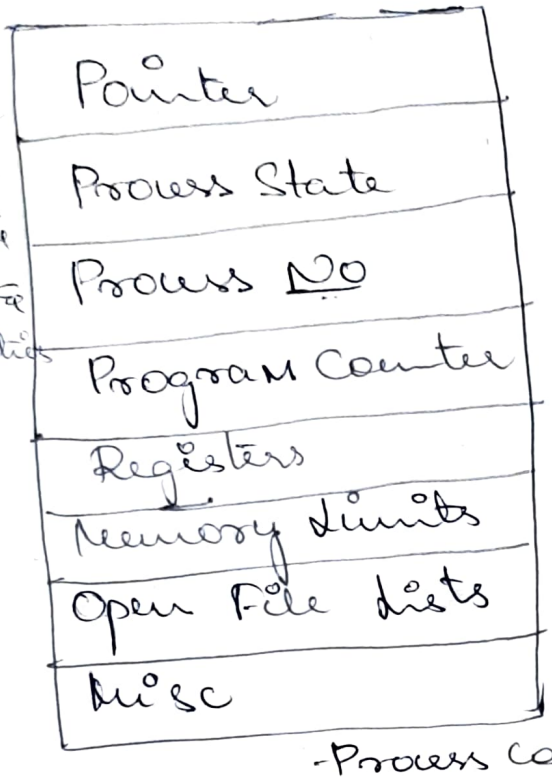
If the process is intensive in terms of IO operations then it is called IO Bound Process.

Process table and Process Control Block.

A process control Block (PCB) contains info. about the process i.e registers, time quantum, priority etc.

The process table is an array of PCB's, that means logically contains a PCB for all of the current processes in the system.

-> PT is used for context switching & scheduling & other activities



- Process control Block

PCB is a Data structure used by O.S to store all info. related to process.
also known as process descripter.
When a process is created the O.S creates corresponding PCB

fig: Process table & PCB.