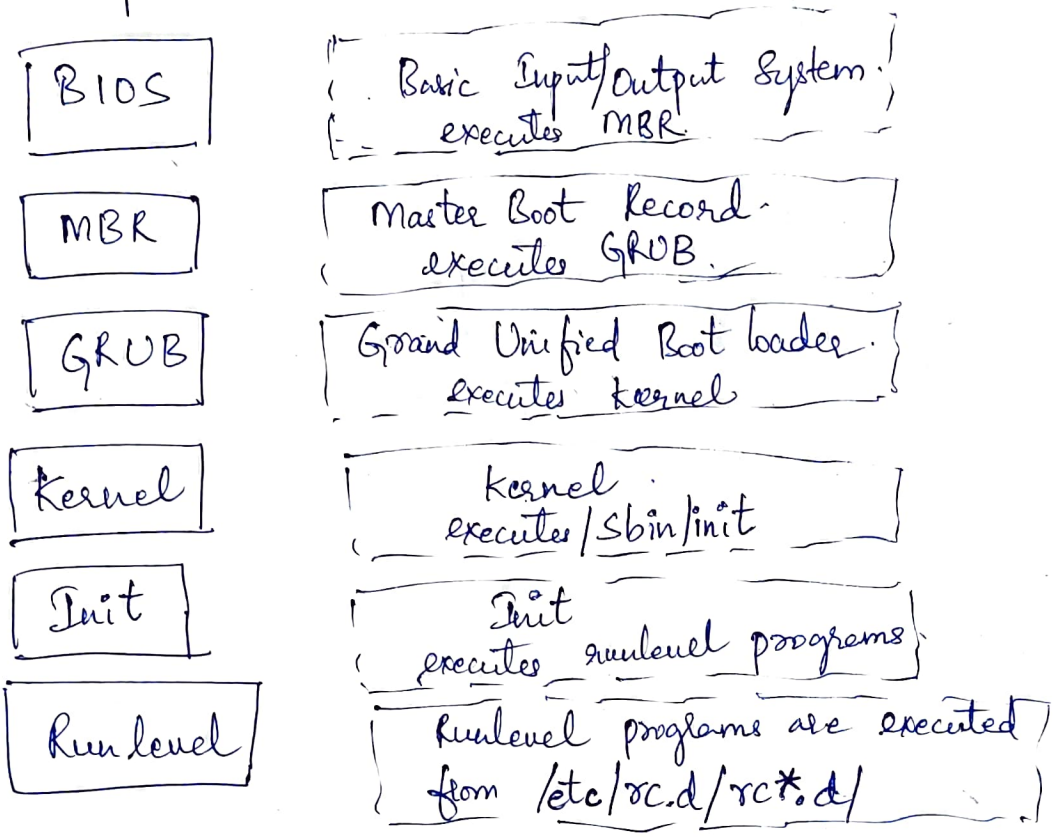
① Linux Boot process (Startup Sequence).

The following are the 6 high level stages of typical Linux boot process.



BIOS : 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 (typically F12 or F2, but it depends on your system) during the BIOS startup 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 :- Master Boot Record.

→ It is located in the 1st sector of the bootable disk. Typically /dev/hda, (or) /dev/sda

→ MBR is less than 512 bytes in size. This has three 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.
4) It contains information about GRUB (or LILO in old systems)
5) So, in simple terms MBR loads and executes the GRUB boot loader.

GRUB :- Grand Unified Bootloader.

→ If you have multiple kernel 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 (the older Linux loader LILO didn't understand filesystem)

→ GRUB configuration file is /boot/grub/grub.conf (/etc/grub.conf is a link to this).

The following is sample grub.conf of CentOS.

```
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title Centos (2.6.18-194.el5PAE)
        root (hd0,0)
        kernel /boot/vmlinuz-2.6.18-194.el5PAE ro root=LABEL=/
        initrd /boot/initrd-2.6.18-194.el5PAE.img
```

→ As you notice the sample code, it contains kernel and initrd image.

→ So, in simple terms GRUB just loads and executes kernel and initrd images.

Kernel

→ Mounts 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 a 'ps -ef | grep init' and check the pid.

→ initrd stands for Initial RAM Disk

→ initrd is used by kernel as temporary root file system until kernel 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 partitions and other hardware.

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 initlevel from ~~/etc~~ /etc/inittab and uses that to load all appropriate programs

→ Execute 'grep 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.

Run level Programs

→ When the linux system is booting up, you might see various services getting started. For example, 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, the system will execute the programs from one of the following directories

(i) Run level 0 - /etc/rc.d/rc0.d/
(ii) Run level 1 - /etc/rc.d/rc1.d/
(iii) Run level 2 - /etc/rc.d/rc2.d/
(iv) Run level 3 - /etc/rc.d/rc3.d/
(v) Run level 4 - /etc/rc.d/rc4.d/
(vi) Run level 5 - /etc/rc.d/rc5.d/
(vii) Run level 6 - /etc/rc.d/rc6.d/

→ Please note that there are also symbolic links available for these 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

→ Program starts with S are used during startup. S for startup.

→ Program starts with K are used during shutdown. K for kill

→ There are numbers right next to S and K in the program names. Those are the sequence number in which the programs should be started or killed

For example; S12syslog is to start the syslog deamon, which has the sequence number of 12. S80sendmail is to start the sendmail daemon, which has the sequence number of 80. So, syslog program will be started before sendmail

(2) What is kernel? Different types of kernel?

Kernel is central Component of an operating system that manages operations of Computer and hardware. It basically manages operations of memory and CPU time. It is core-Component of an operating system. Kernel acts as a bridge to applications and data processing performed at hardware level using inter-process Communication & System Calls

① Monolithic kernel

It is one of types of kernel where all operating system services operate in kernel space. It has dependencies b/w system components. It has huge lines of code which is complex.

Example: UNIX, LINUX, Open VMS, XTS-400 etc..

Advantages:- It has good performance.

Disadvantages:- It has dependencies b/w system component and lines of code in millions

② Micro kernel

It is kernel types which has minimalist approach. It has virtual memory and thread scheduling. It is more stable with less services in kernel space. It puts rest in user space.

Example:- Mach, L4, AmigaOS, Minix, K42 etc

Advantages:- It is more stable.

Disadvantages:- there are lots of system calls and Context switches.

③ Hybrid kernel

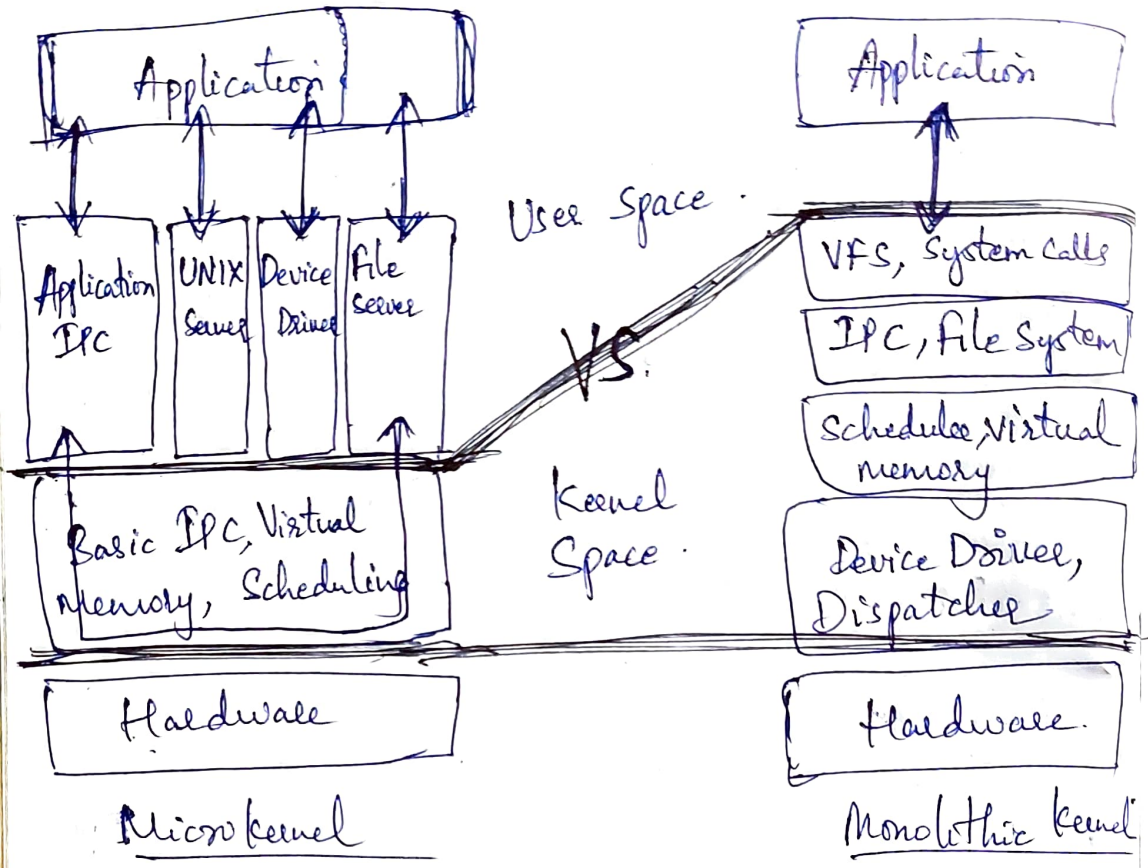
It is combination of both monolithic kernel and microkernel. It has speed and design of monolithic kernel and modularity and stability of microkernel.

Example:- Windows NT, Netware, BeOS etc

Advantages:- It combines both monolithic kernel and microkernel

Disadvta:- It is still similar to monolithic kernel.

③ Difference b/w Microkernel and Monolithic kernel



Basis for Comparsion	Microkernel	Monolithic kernel
1) Basic	In microkernel user services and kernel services are kept in seperate address space.	In monolithic kernel, both user services and kernel services are kept in the same address space
2) Size	Microkernel are smaller in size	Monolithic is larger than microkernel.
3) Execution	Slow execution	Fast execution
4) Extendible	The microkernel is easily extendible	The monolithic kernel is hard to extend
5) Security	If a service crashes, it does effect on working of microkernel	If a service crashes, the whole system crashes in monolithic kernel.
6) Code	To write a microkernel, more code is required	To write a monolithic kernel less code is required
7) Example.	QNX, Symbian, L4Linux, Singularity, K42, Mac OS X, Integrity, Minix, HURD	LINUX, BSDs (FreeBSD, OpenBSD, NetBSD), Microsoft Windows (95, 98, Me) Solaris

(4) Functions of OS :-

OS is a program on which application programs are executed and acts as a communication bridge b/w the user and computer hardware

1. Memory Management

→ Allocates & deallocates memory.
→ Keeps a record of which part of ~~memory~~ primary memory is used by whom & how much
→ Distributes the memory while multiprocessing.

2. Processor Management / Scheduling

→ Allocates and deallocates processor to the processes
→ Keeps record of CPU status.

3. Device management.

→ Allocates and deallocates devices to different processes
→ Keep records of the devices
→ Decides which process can use which device for how much time.

4. File Management

→ keeps records of the status and locations of files
→ Allocates and deallocates resources.

5. Security :

The OS keeps the system and programs safe and secure through authentication. A userid and password decide the authenticity of the user.

6. Accounting

As the operating system keeps track of all the functions of a Computer System. Hence it makes a record of the activities taking place on the system.

7. Other Functions

→ Error detection
→ keeping a record of System performance.
→ Communication b/w different Software etc.

(5) Difference b/w UEFI & Legacy boot.

Legacy boot	UEFI
→ Legacy boot is the boot process used by BIOS firmware	→ UEFI stands for Unified Extensible Firmware Interface.
→ It stores a list of installed storage devices that are bootable according to a configurable order of priority	→ It provides a user-friendly Graphical User Interface and recognises large storage devices as opposed to legacy boot
→ Security and Efficiency is lower compared to UEFI	→ Has additional security features and is more efficient
→ Less User friendly	→ More user friendly.
→ Uses the MBR partition	→ Uses the GPT partitioning scheme.
→ Uses BIOS firmware for boot process.	→ Uses UEFI firmware for boot process.

(6) Discuss on Operating System Linux, Windows & Mac OS

Linux	Windows	Mac OS
→ Open Source.	→ Closed Source	→ Closed Source.
→ Stores data in the form of tree. There is a single file tree and all the drives are mounted on this tree	→ Uses directory structure to store different kinds of files of the user. It has logical drives and Cabinet drawers	→ Uses the file structure Commonly known as Mac OS X
→ Does not have a specific registry of its own	→ Registry is a master database which stores all the settings	→ Stores all application settings in a series of plist files

→ Provides terminal → Terminal is Command prompt. → Provides Console as terminal

→ Easy to switch interfaces → Not interchangable interfaces till windows 8 → Has a facility to bridge virtual network interfaces.

(6) Commands on Windows OS to check disk partition

Step1:- Open Command prompt.

Step2:- Use 'diskpart' Command.

The following Commands can be used to operate on disk/disk Volumes after 'diskpart' Command.

Active :- Mark the selected partition as active
Add :- Add a mirror to a simple volume.
Assign :- Assign a drive letter (or) mount point to the selected.
Attributes :- Manipulates disk/Volume attributes
Break :- Break a mirror set.
Clean :- Clear information off the disk
Compact :- Attempt to reduce physical size of disk/files
Create :- Create a volume, partition or virtual disk
Delete :- Delete an object.
Detail :- Details of an object
Detach :- Detaches a virtual disk file.
Exit :- Exit diskpart
Extend :- Extend a Volume
Expand :- Expands the max size of virtual disk.
Filesystems :- Display of current & supported file system

Format :- Format Volume/partition.
Help :- Display a list of Commands
Import :- Import disk group.
Inactive :- Mark selected as inactive.
List :- Display list of objects.
Merge :- Merges child disks with parent disk
Remove :- Remove a driver letter/mount point assignment
Select :- Shift the focus on an object.

(7) List the Commands to check Services in Windows.

One can open Services from Command prompt by the Command Services.msc and pressing Enter in Keyboard.

(8) List the steps to check disk partitions in windows.

Step1 :- Open file explorer.
Step2 :- Right Click on 'This PC'
Step3 :- Choose 'Manage' from the pop-up menu
Step4 :- Navigate to Storage → Disk Management in navigation panel.

(9) List the steps to start or stop Services in windows

Step1 :- Hit windows key + R to open the run window.
Step2 :- Type in services.msc in the open : box
Step3 :- Services dialog box/window will open
Step4 :- Select the service to start / stop
Step5 :- Choose the relevent option to operate.