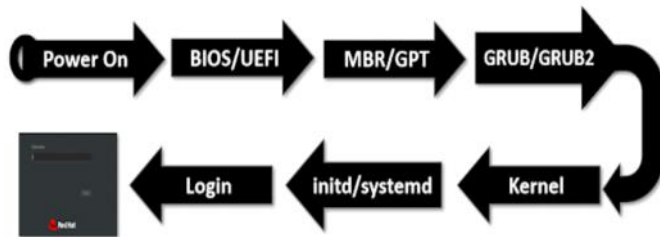


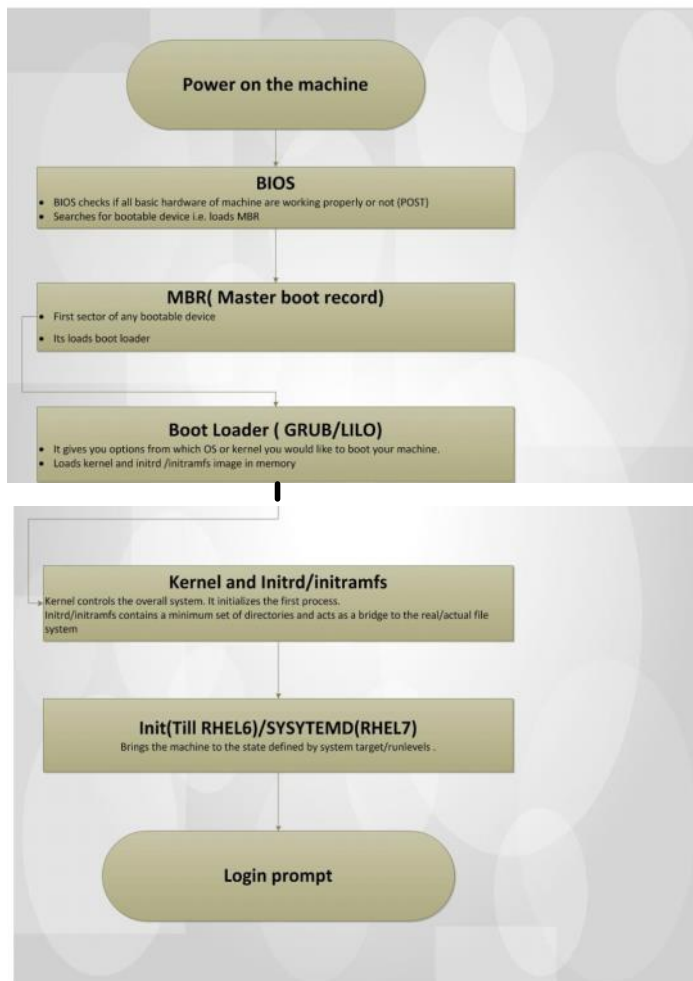
Booting process of Linux Operating System ?

- The boot process is what happens between the moment you power on a machine (physical or virtual) and when the operating system has been completely loaded.
- The boot process is made up of a series of necessary actions that a computer goes through upon startup.
- By understanding the Linux boot process, you can quickly diagnose and troubleshoot what is wrong with your system and then fix it.

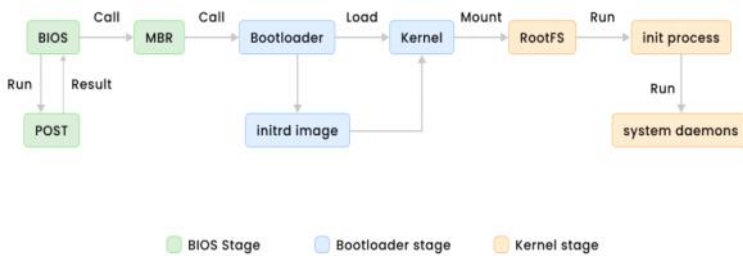
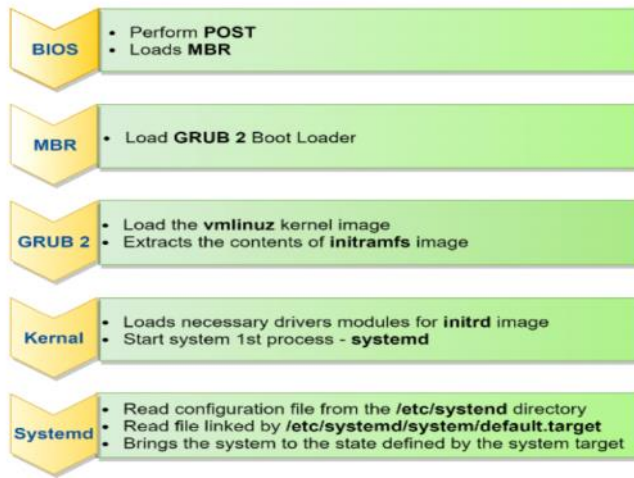
First Image to Understand About the Booting Process ?



Second Image to Understand About the Booting Process ?



Third and Forth Image to Understand About the Booting Process ?



1- Power ON

- The Machine is Powered on.
- SMPS is an electronic power supply system that makes use of a switching regulator to transfer electrical power effectively.
- It is a PSU (power supply unit) and is usually used in computers to change the voltage to the appropriate range for the computer.
- A power supply unit (PSU) converts mains AC to low-voltage regulated DC power for the internal components of a desktop computer.
- The CMOS (complementary metal oxide semiconductor) battery, also called coin-cell battery on the motherboard helps the BIOS or UEFI to store the hardware configuration settings. It helps keep a record of the date and time when the computer is switched off.

2- BIOS/UEFI

- BIOS stands for Basic Input/Output System.
- It is a program that is stored in read-only memory chips on the motherboard of the computer.
- As soon as our computer is ON, this is the program that gets executed first.
- It runs regardless of what OS you have installed on your machine. It is not associated with any Operating system.
- Once this program is executed, it checks if the basic hardware of machine which is required to start the machine is working properly or not. Like the keyboard, RAM, visual display unit etc.
- This process of checking basic hardware components if they are working fine or not is known as a POST(power on self-test).

Note: The System Firmware, either modern UEFI OR older BIOS, runs a Power on Self-test (POST) and starts to initialize some of the hardware.

Note: Bios Setup OR BIOS Settings can be changed by pressing F12 or F2 it depends on your H/W vendor Settings

- BIOS/UEFI locates the boot device & loads MBR/GPT (GUID Partition Table) from an active partition.
OR
After POST BIOS checks for a bootable device. From where, OS can be booted. The bootable device can be anything like hard disk, floppy disk, pen drive.
- Once BIOS finds the bootable device it hands over control to the first sector of the bootable device, **which is MBR**.
- If in case of BIOS does not find any bootable device then BIOS gives an error and machine boot process is stopped.

Important Points:

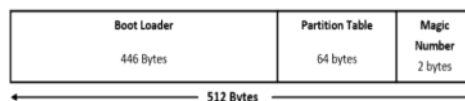
- Nowadays, UEFI (Unified Extensible Firmware Interface) is used in place of BIOS, which has better features such as faster boot time, more security, hardware size more than 2TB can be supported.
- There is no way to switch from BIOS to UEFI, you need to buy new hardware that includes UEFI, as most of the machine now comes with UEFI. UEFI supports GPT partition instead of MBR.

UEFI vs BIOS: Comparison Chart

Feature Category	BIOS	UEFI
Release Date	1981	2002 (first specification released)
User Interface	Text-based, keyboard navigation	Graphical, supports mouse and keyboard
Operating Mode	16-bit, limited to 1MB of addressable space	32-bit or 64-bit, can access more memory
Partition Support	MBR (Master Boot Record), up to 2TB	GPT (GUID Partition Table), over 2TB
Security	Basic, no inherent security feature	Supports Secure Boot, prevents unauthorized OS
Performance	Slower boot times, limited hardware support	Faster boot times, optimized for modern hardware

	MBR	GPT
Maximum Partition Capacity	2TB	9.4ZB (1 ZB is 1 billion terabytes)
Maximum Partition Number	4 primary partitions(or 3 primary + an infinite number of logical partitions)	128 primary partitions
Firmware Interface Support	BIOS	UEFI
Operating System Support	Windows 7 and older systems like Windows 95/98, Windows XP 32-bit, Windows 2000, Windows 2003 32-bit	Later systems like Windows 11, Windows 10 64-bit, Windows 8/8.1 64-bit

3- MBR [Master Boot Record]



- MBR stands for Master Boot Record.
- It contains the information about Boot Loader of the Current installed Operating System.
- So in simple words, MBR loads and execute the **GRUB boot loader**.

Note:-

Its size is 512 bytes. It is the first sector of any bootable device.

It is divided into 3 parts.

- 1- Primary Boot loader information in first 446 Bytes.
- 2- Partition Table information in next 64 Bytes.
- 3- MBR Validation Check in last 2 Bytes.

4- GRUB [Grand Unified Boot loader]

- Its main job of Bootloader is to load **kernel and initrd /initramfs image** in memory.
- Once boot loader loads kernel in RAM it passes control to **Kernel**.
- It gives you options from which OS or kernel you would like to boot your machine
- So in simple words, GRUB OR Bootloader just load and executes **Kernel and Initramfs (initrd) images** and give control to **KERNEL**.

Note:-

- The default boot loader is GRUB-2 (Grand Unified Bootloader Version-2) is available in RHEL-7-RHE-8-and-RHEL-9 OS.
- GRUB2 loads its configuration from the **/boot/grub2/grub.cfg** file and displays a menu where you can select which kernel to boot. Also read the GRUB menu Settings from

/etc/default/grub and /etc/grub.d directory.

- After you select a kernel, or the timeout expires, the boot loader loads the kernel and initramfs from disk and places them in memory.



What is the Use of Kernel and Initramfs OR What is the Relation between kernel and initramfs ?

- The kernel manages memory, input-output device. In short, it controls the overall system. You can say heart of Linux OS.
- Kernel is compiled with different drivers but they are not able to load actual ROOT file system because some additional drivers are required for it which is available in initramfs image.
- initrd or initramfs is used by kernel as temporary root file system until kernel is booted and the real root file system is mounted.
- An initramfs is an archive containing the kernel modules for all the hardware required at boot, initialization scripts, minimum set of directories and more to mount the real root file system.
- It detects what all device drivers are needed to load the actual file system and it loads them from a temporary file system. After that, other partition like Partition, SWAP, LVM, RAID etc is mounted then initrd is unmounted.
- We can see kernel and initramfs file in /boot directory. Starting with vmlinuz denotes kernel.
- Initially, the kernel is loaded in read-only mode.
- Initramfs is used by 2.6 kernels whereas initrd was used by older 2.4 (and earlier) kernels.

```
[root@server ~]# ls /boot
config-3.10.0-327.el7.x86_64
grub2
initramfs-0-rescue-e0bdfb7f861a4aa69db9ca2e34abfa03.img
initramfs-3.10.0-327.el7.x86_64.img
initramfs-3.10.0-327.el7.x86_64kdump.img
initrd-plymouth.img
symvers-3.10.0-327.el7.x86_64.gz
System.map-3.10.0-327.el7.x86_64
vmlinuz-0-rescue-e0bdfb7f861a4aa69db9ca2e34abfa03
vmlinuz-3.10.0-327.el7.x86_64
```

5- KERNEL

- Once the real file system is mounted, then kernel runs the first process i.e init(till RHEL 6) / Systemd from (RHEL 7).
- The kernel initializes all hardware for which it can find a driver in the initramfs, then executes /sbin/init from the initramfs as PID is 1.
- In Red Hat Enterprise Linux 7-8-9, /sbin/init is a link to systemd.

In simple terms, Bootloader gives control to Kernel and Now kernel will give control to Systemd.

6- SYSTEMD

Note:-

systemd uses targets instead of runlevels. inittab is no longer used when using systemd.

By default, there are two main targets:

- a) multi-user.target: analogous to runlevel 3

b) graphical.target: analogous to runlevel 5

- Systemd read all the information from using `/etc/systemd/system/default.target` and `/etc/systemd/system/`.
- The systemd executes all units for the `initrd.target` target.
- This includes mounting the root file system on disk on to the `/sysroot` directory.
- Configured using `/etc/fstab`
- The kernel switches the root file system from `initramfs` to the root file system in `/sysroot`. systemd then re-executes itself using the copy of systemd installed on the disk.
- systemd looks for a default target, either passed in from the kernel command line or configured on the system, then starts (and stops) units to comply with the configuration for that target, solving dependencies between units automatically.
- In essence, a systemd target is a set of units that the system should activate to reach the desired state.
- These targets typically start a text-based login or a graphical login screen.

=====

7- Login Screen

Once all required services and script are executed user gets login prompt.

Username:
password:

Here It will verify the all information :

- `/etc/passwd`
- `/etc/group`
- `/etc/shadow`
- Password policy
- All bash profiles files of user.
- Then you will get command prompt as per defined Shell.
- If all things are ok then you will command a command prompt as per user types.

[root@ localhost ~] # OR [krishna @ localhost ~] \$

=====