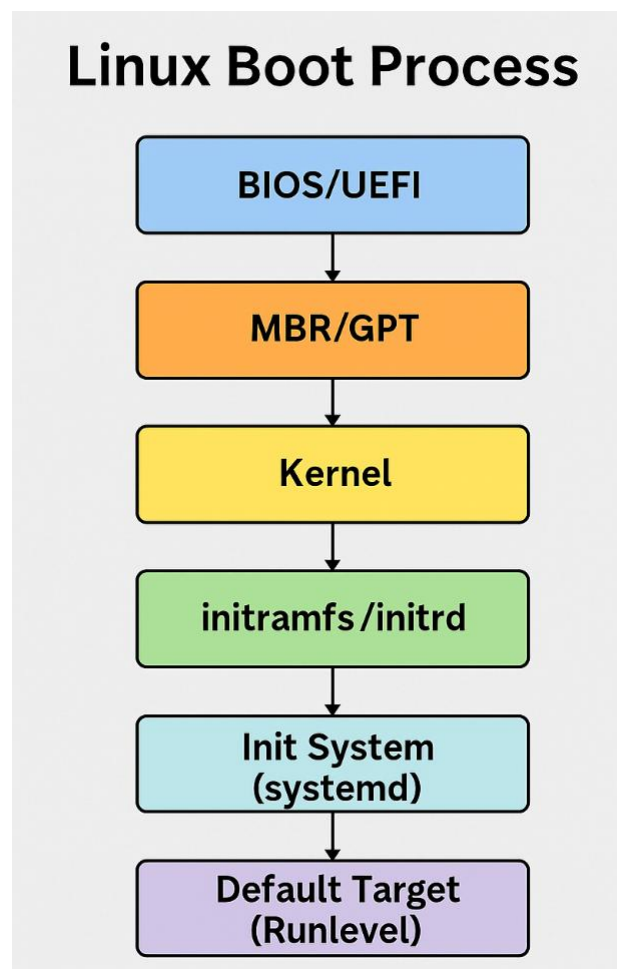


The Linux Boot Process — Explained Step-by-Step

Understanding the Linux boot process is essential for system administrators, DevOps engineers, and anyone managing Linux systems. From powering on your server to reaching the login screen, the system follows a precise set of steps.

Main steps in the Linux Boot Process

1. BIOS/UEFI initialization
2. MBR/GPT Execution
3. GRUB (Bootloader)
4. Kernel Loading
5. Init system (systemd)
6. Default target (Runlevel)



1. BIOS / UEFI – System Startup

When you power on the machine, it begins in **BIOS** (older) or **UEFI** (modern systems) phase. Then the hardware performs a POST (Power-On Self Test), after which the BIOS or UEFI firmware takes over.

UEFI stands for **Unified Extensible Firmware Interface**.

BIOS stands for **Basic Input/Output System**.

- If the system is using **MBR** (Master Boot Record), the BIOS reads the MBR from the first sector of the disk to find the bootloader.
- If the system is using **GPT** (GUID Partition Table), UEFI firmware will read the GPT partition table and look for the EFI System Partition (ESP), where the bootloader is stored.

Note : If there is any error, then it will return the error code (series of beeps)

2. MBR / GPT – Partition Table Lookup

MBR (Master Boot Record) is a legacy boot method; BIOS reads the first 512 bytes of the disk.

GPT (GUID Partition Table) is newer; UEFI reads the **EFI System Partition (ESP)** to locate the bootloader.

*To check the partition scheme : **sudo parted -l***

3. GRUB- The Bootloader

GRUB2 (Grand Unified Bootloader v2) is loaded by MBR or UEFI. It loads the **Linux kernel** into memory.

If multiple kernels are installed, it shows a boot menu. It displays those on a splash screen and wait for a few sec to select, if not selected then it will select the default one which is mentioned in the grub configuration file.

*Grub configuratin file path : **/boot/grub2/grub.cfg***

4. Kernel - The Core of OS

The Linux Kernel initializes the drivers, CPU and Memory.

It loads the drivers from the **initrd.img (Ubuntu Linux)** and **initramfs-4 (RHEL, Oracle Linux)**, It mounts the /root file system, and initializes the 1st system process **systemd (PID 1)** - system daemon

initrd/initramfs-- used as temp root file system which was mounted at the early stages of boot process, which helps to load the other drivers and kernel modules.

System stills looks for /sbin/init as the entry point but which is symlink to systemd

/sbin/init → symlink → /lib/systemd/systemd

5. systemd - The first process

systemd is the current standard for the service management and system initialization.

systemd then starts the all the required process and reads the default.target file to find the runlevel to bring the system to that runlevel.

Default target file path : **/etc/systemd/system/default.target**

6. Runlevel / Target – Final Boot State

systemd uses **target** not runlevels, however they serve the same purpose. Each target defines what services and units are activated.

Runlevel	Purpose	systemd Target
0	Halt / Shutdown	poweroff.target
1	Single-user mode	rescue.target
3	Multi-user (CLI only)	multi-user.target
5	Multi-user + GUI	graphical.target
6	Reboot	reboot.target