

Linux Boot Process

BIOS:

- Stands for Basic I/O system.
- First Program that is executed which is stored in read-only memory on the motherboard of the computer.
- Perform POST (power-on self-test) verify the hardware components and peripheral to ensure if the computer is in working condition.
- Check for bootable devices like Hard disk, Pen drive, CD.
- Once a bootable device is detected it will handover control to the first sector of the bootable device. I.e., MBR.
- Apart from BIOS, UEFI (Unified Extensible Firmware Interface) is also used.

MBR:

- Stand for Master Boot Record.
- It's 512 bytes in size, the first sector of any bootable device contains machine code instructions to boot a machine and having following info.
 - Boot Loader (446 bytes)
 - Partition Table (64 bytes)
 - Error Checking (2 bytes)
- It will load the boot loader into the memory and hand it over to it.

GRUB:

- Load `/boot/grub2/grub.cfg` at boot time.
- At this stage, the user will see GUI asking different OS or Kernels configured to boot.
- Once you selected the kernel, it locates the corresponding kernel binary `/boot/vmlinuz-<kernel-version>`
- Main job is to load the kernel and initrd/initramfs image into memory.
- Note: initrd used before Linux 7 from Linux 7 initramfs is running.
- Once Kernel loads in RAM, it passes control on it.

IN RHEL7

Default boot loader in GRUB2

GRUB is for X86 architecture, it could be different for other arch like for Intel Itanium - ELILO

Kernel:

- First kernel is loaded into memory into read-only mode.
- Initramfs/Initrd gets decompressed and then First loads a temporary root file system.
- Initrd then detects and loads the device from the temporary file system to load the actual file system.
- Mount other partitions like LVM, RAID etc and unmount itself.
- Once the root filesystem is mounted, the kernel initialises the first process init/systemd.

Kernel and Initramfs file will be stored in the `/boot` folder.

Systemd:

- First service started with process ID 1 by kernel.
- Start all required processes `/etc/systemd/system/default.target`.
- To bring the system to the run-level/target (0-6)
- We can find the different run levels files under
 - `/usr/lib/systemd/system`
 - `ls -l runlevels*`

Run Levels

- Init 0 - Shutdown
- Init 1 - Single User Mode
- Init 2 - Multi User without Network
- Init 3 - Multi User with Network
- Init 4 - Unused
- Init 5 - GUI Mode
- Init 6 - Restart

Common Issues

The Motherboard firmware, also known as BIOS or UEFI, is a type of software's stored in a non-volatile memory chip on the motherboard. It provides essential functionality for the computer's hardware, including:

1. Initialization: The firmware initialises hardware components such as CPU, Memory, Storage devices, and peripherals during the boot process.
2. System Configuration: It stores configuration settings for the hardware, including settings related to boot order, CPU parameters, memory things and other system settings.

Notes: The firmware type (BIOS/UEFI) is determined by the hardware design and firmware chip installed on the motherboard.

--

What happens if `/boot/grub2/grub.cfg` files are deleted or corrupted.

1. Boot Failure: When we restart the computer, GRUB will attempt to load the configuration file. Since the file is missing or corrupted, GRUB will encounter an error and be unable to proceed with the boot process.
2. Inability to Boot: Without proper bootloader configuration file, GRUB won't have the necessary information to locate and load the kernel, `initramfs`, and other essentials files required for the booting the OS.
3. GRUB Rescue Mode: In response to the missing `grub.cfg` file, GRUB may enter into "GRUB Rescue Mode." This mode provides a minimal CLI where you can attempt to manually recover or repair the bootloader configuration.

Steps to recover the `grub.cfg` in rescue mode.

- Mount the disk for boot.
 - Exit grub console
 - Go to Troubleshooting
 - Choose Rescue a CentOS System.
 - Then choose 1 to continue.
 - Then hit enter
 - Change the root file system chroot /mnt/sysimage
 - grub2-install /dev/sda
 - grub2-mkconfig -o /boot/grub2/grub.cfg
 - Exit and Exit
-