

## Linux Boot Process - Step by Step Procedure

The stages involved in Linux Boot Process are:

1. BIOS
2. Boot Loader
  - MBR
  - GRUB
3. Kernel
4. Init
5. Runlevel scripts
6. User Interface

### STEP 1: BIOS

This is the first process begins once you power on your machine.

When you press the power button of the machine, CPU looks out into ROM for further instruction.

BIOS determine all the list of bootable devices available in the system.

Prompts to select bootable device which might be Hard Disk, CD/DVD-ROM, Floppy Drive, USB Flash Memory Stick etc..

Operating System tries to boot from Hard Disk where the MBR contains primary boot loader.

### STEP 2: Boot Loader

To be very brief this phase includes loading of the boot loader (MBR and GRUB/LILO) into memory to bring up the kernel.

MBR (Master Boot Record)

It is the first sector of the Hard Disk with a size of 512 bytes.

The first 434 - 446 bytes are the primary boot loader, 64 bytes for partition table and 6 bytes for MBR validation timestamp.

NOTE: Now MBR directly cannot load the kernel as it is unaware of the file system concept and requires a boot loader with file system driver for each supported file systems, so that they can be understood and accessed by the boot loader itself.

To overcome this situation GRUB is used with the details of the file system in `/boot/grub.conf` and file system drivers

GRUB (Grand Unified Boot loader)

this loads the kernel in 3 stages

### **GRUB stage 1:**

The primary boot loader takes up less than 512 bytes of disk space in the MBR - too small a space to contain the instructions necessary to load a complex operating system.

Instead the primary boot loader performs the function of loading either the stage 1.5 or stage 2 boot loader.

### **GRUB Stage 1.5:**

Stage 1 can load the stage 2 directly, but it is normally set up to load the stage 1.5.

This can happen when the `/boot` partition is situated beyond the 1024 cylinder head of the hard drive.

GRUB Stage 1.5 is located in the first 30 KB of Hard Disk immediately after MBR and before the first partition.

This space is utilized to store file system drivers and modules.

This enabled stage 1.5 to load stage 2 to load from any known location on the file system i.e. `/boot/grub`

### **GRUB Stage 2:**

This is responsible for loading kernel from `/boot/grub/grub.conf` and any other modules needed

Loads a GUI interface i.e. splash image located at `/grub/splash.xpm.gz` with list of available kernels where you can manually select the kernel or else after the default timeout value the selected kernel will boot.

The original file is `/etc/grub.conf` of which you can observe a symlink file at `/boot/grub/grub.conf`

### STEP 3. Kernel

This is the heart of operating system responsible for handling all system processes.

#### **Kernel is loaded in the following stages:**

Kernel as soon as it is loaded configures hardware and memory allocated to the system.

Next it uncompresses the `initrd` image (compressed using `zlib` into `zImage` or `bzImage` formats) and mounts it and loads all the necessary drivers.

Loading and unloading of kernel modules is done with the help of programs like `insmod`, and `rmmod` present in the `initrd` image.

Looks out for hard disk types be it a LVM or RAID.

Unmounts `initrd` image and frees up all the memory occupied by the disk image.

Then kernel mounts the root partition as specified in `grub.conf` as read-only.

Next it runs the `init` process

#### **STEP 4: Init Process**

Executes the system to boot into the run level as specified in `/etc/inittab`.

You can check current runlevel details of your system using below command on the terminal

```
# who -r
```

```
run-level 3 Jan 28 23:29 last=S
```

Next as per the `fstab` entry file system's integrity is checked and root partition is re-mounted as read-write (earlier it was mounted as read-only).

#### **STEP 5: Runlevel scripts**

A no. of runlevel scripts are defined inside `/etc/rc.d/rcx.d`

Based on the selected runlevel, the init process then executes startup scripts located in subdirectories of the /etc/rc.d directory.

Scripts used for runlevels 0 to 6 are located in subdirectories /etc/rc.d/rc0.d through /etc/rc.d/rc6.d, respectively.

s and k scripts in the etc rcx.d directories, where the files beginning with the letters "s" is startup scripts and "k" is stopping scripts.

Lastly, init runs whatever it finds in /etc/rc.d/rc.local (regardless of run level). rc.local is rather special in that it is executed every time that you change run levels.

NOTE: rc.local is not used in all the distros as for example Debian.

Next if everything goes fine you should be able to see the Login Screen on your system.

