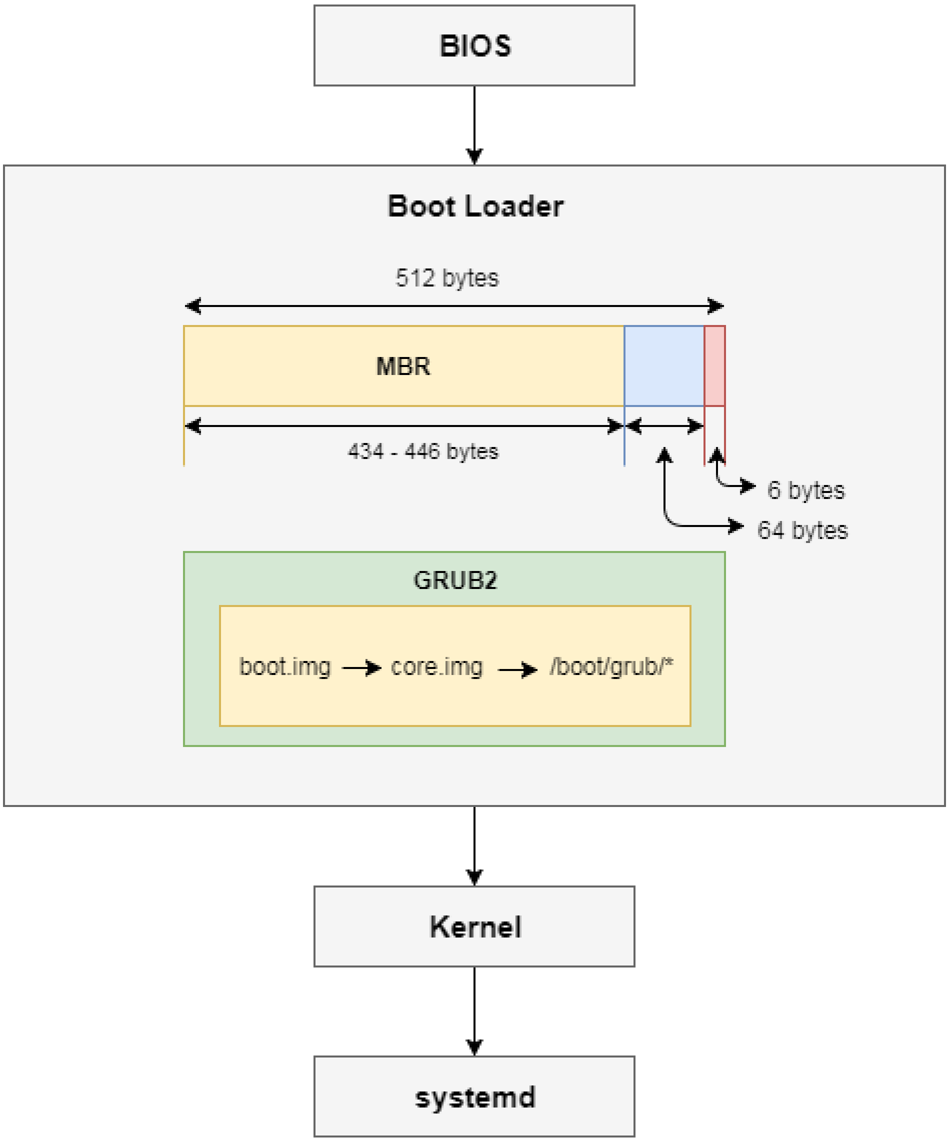
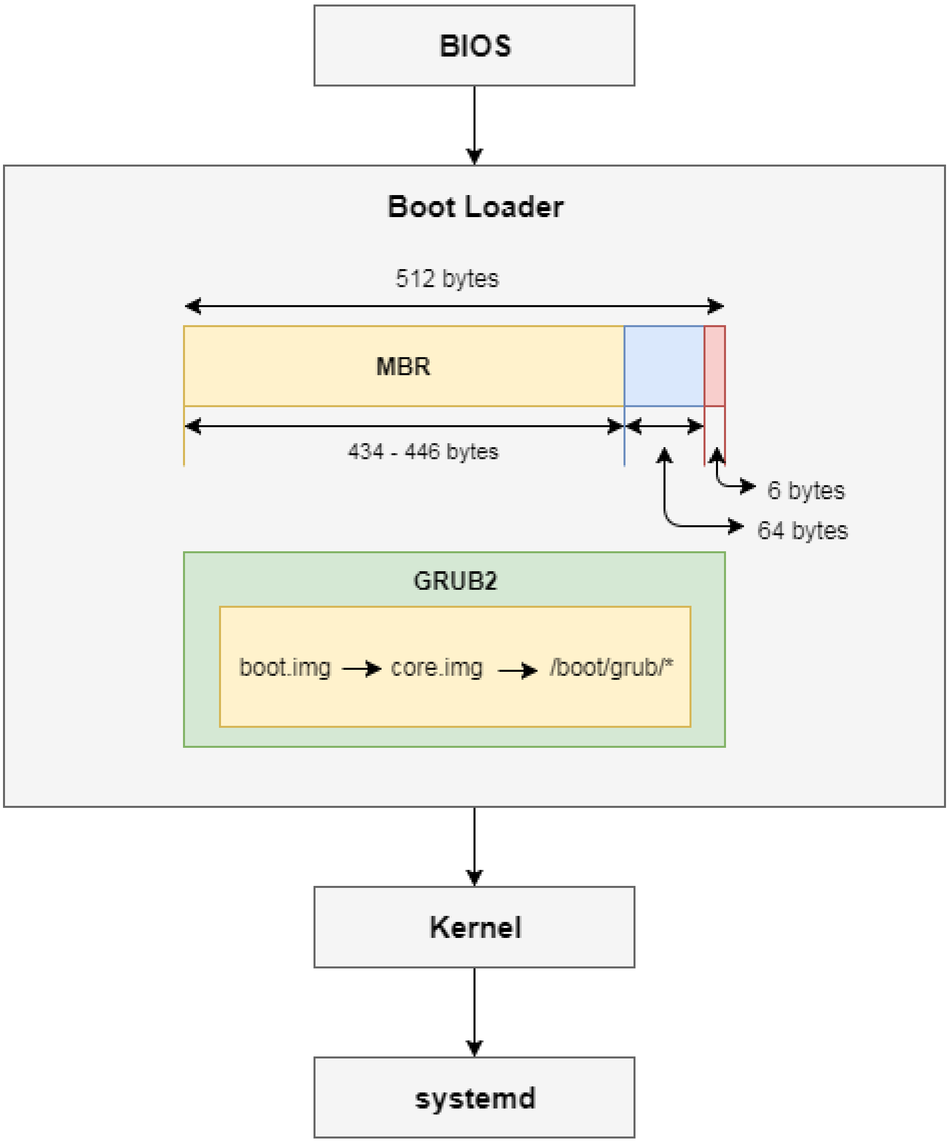
GOLINUXHUB
Sys Admin Tutorial Guide

Menu



Step by Step Linux boot process with GRUB2 and systemd in RHEL 7 / CentOS 7

Below table depicts the flow of the boot up process for RHEL 7 / Cent OS 7 with GRUB2 and systemd



The stages involved in Linux Booting Process are:

BIOS

Boot Loader

– **MBR**

– **GRUB2**

Kernel

systemd

I have also written another article with the steps of boot process using a **flow chart** to help you understand better.

BIOS

- This is the first thing which loads once you power on your machine.
- When you press the power button of the machine, CPU looks out into ROM for further instruction.
- The ROM contains JUMP function in the form of instrucion which tells the CPU to b up the BIOS

- BIOS determines all the list of bootable devices available in the system.
- Prompts to select bootable device which can be Hard Disk, CD/DVD-ROM, Floppy Drive, USB Flash Memory Stick etc (optional)
- Operating System tries to boot from Hard Disk where the MBR contains primary boot loader.

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 filesystem 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 GRUB2 is used with the details of the filesystem in **/boot/grub2/grub.cfg** and file system drivers.

GRUB2

- Its function is to take over from BIOS at boot time, => then load itself, => load the Linux kernel into memory, => and then turn over execution to the kernel.
- Once the kernel takes over, => GRUB has done its job and it is no longer needed.
- GRUB supports multiple Linux kernels and allows the user to select between them at boot time using a menu.
- So you can choose between various kernel or failover kernel if the primary one does not works.
- The **/boot/grub2/grub.cfg** file is the GRUB configuration file.
- It is generated by the **grub2-mkconfig** program using a set of primary configuration files and the grub default file as a source for user configuration specifications from **/etc/sysconfig/grub**.
- The **/boot/grub2/grub.cfg** file is first generated during Linux installation and regenerated when a new kernel is installed.

To check the GRUB2 version used

```
# grub2-install -V
grub2-install (GRUB) 2.02~beta2
```

With legacy GRUB we had thre different stages i.e.

Stage 1 -> Stage 1.5 -> Stage 2

But with GRUB2 these stages do not exist anymore instead GRUB2 contain various *.img files as explained below

boot.img

On PC BIOS systems, this image is the first part of GRUB to start. It is written to a master boot record (MBR) or to the boot sector of a partition. Because a PC boot sector is 512 bytes, the size of this image is exactly 512 bytes.

The sole function of boot.img is to read the first sector of the core image from a local disk and jump to it. Because of the size restriction, boot.img cannot understand any file system structure, so grub-install hardcodes the location of the first sector of the core image into boot.img when installing GRUB.

core.img

This is the core image of GRUB. It is built dynamically from the kernel image and an arbitrary list of modules by the grub-mkimage program. Usually, it contains enough modules to access /boot/grub, and loads everything else (including menu handling, the ability to load target operating systems, and so on) from the file system at run-time. The modular design allows the core image to be kept small, since the areas of disk where it must be installed are often as small as 32KB.

Once executed, core.img will load its configuration file and any other modules needed, particularly file system drivers at installation time based on the location using which the system is getting booted as explained below

- **diskboot.img**
  This image is used as the first sector of the core image when booting from a hard disk. It reads the rest of the core image into memory and starts the kernel. Since file system handling is not yet available, it encodes the location of the core image using a block list format.

- **cdboot.img**
  This image is used as the first sector of the core image when booting from a CD-ROM drive. It performs a similar function to diskboot.img.

- **pxeboot.img**
  This image is used as the start of the core image when booting from the network using PXE. See Network.

- **lnxboot.img**
  This image may be placed at the start of the core image in order to make GRUB look enough like a Linux kernel that it can be booted by LILO using an 'image=' section.

- **kernel.img**
  This image contains GRUB's basic run-time facilities: frameworks for device and file handling, environment variables, the rescue mode command-line parser, and so on. It is rarely used directly, but is built into all core images.

Last Stage

- At this point GRUB2 loads the selected kernel into memory and turns control of the computer over to the kernel.
- If files or the partition become unavailable, or if the user wishes to take direct control, stage 2 will drop the user to the GRUB command prompt, where the user can then manually specify the boot parameters.

Kernel

- The kernel files are present inside /boot directory along with an initial RAM disk image, and device maps of the hard drives.

```
 # ls -l /boot/
total 88757
-rw-r–r–. 1 root root   140898 Oct 13 20:21 config-3.10.0-693.5.2.el7.x86_64
drwxr-xr-x. 3 root root     1024 Dec 20 15:16 efi
drwx——. 5 root root     1024 Dec 20 15:30 grub2
-rw——-. 1 root root 54730516 Dec 20 15:20 initramfs-0-rescue-0c042a4bdcc7466bb05a0fd73faafc74.img
-rw——-  1 root root 20098955 Dec 20 15:30 initramfs-3.10.0-693.5.2.el7.x86_64.img
-rw-r–r–. 1 root root   611512 Dec 20 15:18 initrd-plymouth.img
drwx——. 2 root root    12288 Dec 20 15:16 lost+found
-rw-r–r–. 1 root root   293084 Oct 13 20:22 symvers-3.10.0-693.5.2.el7.x86_64.gz
-rw——-. 1 root root  3228852 Oct 13 20:21 System.map-3.10.0-693.5.2.el7.x86_64
-rwxr-xr-x. 1 root root  5877008 Dec 20 15:20 vmlinuz-0-rescue-0c042a4bdcc7466bb05a0fd73faafc74
-rwxr-xr-x. 1 root root  5877008 Oct 13 20:21 vmlinuz-3.10.0-693.5.2.el7.x86_64
```

- As you see it contains the vmlinuz and initrd files for rescue and the default kernel which will be loaded if nothing else is selected in the GRUB menu

- The initrd file contains a minimal set of directories and executables to achieve this, such as the insmod tool to install kernel modules into the kernel. In other words, it contains the necessary executables and system files to support the second-stage boot of a Linux system.
- Here the initramfs file extracts itselfs and loads systemd

Systemd

- systemd then is responsible for probing all remaining hardware, mounting all necessary file systems and spawning all configured services
- To brief the steps involved systemd will load the filesystem based on default.target file. A long listing of the default.target file shows you which target starts when the system boots

```
[root@golinuxhub ~]# cd /etc/systemd/system/

[root@golinuxhub system]# ls -l default.target
lrwxrwxrwx. 1 root root 41 Dec 24 23:31 default.target -> /usr/lib/systemd/system/multi-user.target
```

- Since the default.target is linked to multi-user.target the systemd will start all the necessary services enabled for this target level but this is all done in the below order

local-fs-pre.target -> sysinit.target -> basic.target -> multi-user.target
This could be summarised in the below flow of tasks

- Import the network configuration from the initramfs
- run a file system check (fsck) on the root file system when necessary, and remounting the root file system (and special kernel file systems) based on the contents of the /etc/fstab file.
- Set kernel options based on the sysctl values
- Start the udevd daemon to detect hardware and run various sysctl related params
- Decrypt the filesystem if encrypted
- Mount the filesystems
- Enabling swap devices
- Start all the services enabled for multi-user.target
- If graphical.target is the default target then systemd will start the display-manager.service which runs the GNOME display manager.

References:

GNU GRUB Manual

GNU GRUB Wiki

‹ How do I set or change (default) runlevel using systemd in CentOS 7/RHEL 7

› Tutorial / Cheatsheet: 11 examples to use systemctl to manage unit files and service in RHEL 7 / CentOS 7

2 thoughts on “Step by Step Linux boot process with GRUB2 and systemd in RHEL 7 / CentOS 7”

unknown

april 21, 2018 at 9:12 pm

Thanks, Good Post

Reply

unknown

november 20, 2018 at 12:10 pm

nice

Reply

Leave a Comment

Name *

Email *

Website

□ Save my name, email, and website in this browser for the next time I comment.

Post Comment

Search …

Also Read

Openswan configuration on Red Hat5 for Sonicwall

How to configure Squirrelmal in Linux

How to configure local customised repository for zypper based installation in SuSE Enterprise Linux

Pros and Cons of VPN Gambling

Part 2: Configure Openstack OVSBridge, Network (Neutron), Public and Private Network, Router in Openstack

How to create a user (normal and root) using kickstart configuration file in RHEL 7 / CentC

iptables rules for Samba 4 in Red Hat Linux

How to create a password less ssh connection in Linux

How to change the MAC Address of Ethernet device in Linux?

How to Fix " KERBEROS_V4 rejected as an authentication type"

Latest Articles

Smart Home Devices Every Individual Must Have

What is a VPN and Why You May Need One

Mobile Casinos: Then and Now

The best screen recorders in 2021. What will be popular in 2022?

Benefits of Choosing an Unlimited data Plan

Top Most Used and Popular Mobile Apps in 2021

Top 7 Linux Programming Books

Linux Distros: Which is the Best for Privacy and Security?

Pros and Cons of VPN Gambling

How Advanced Technology Enhances Digital Marketing Effectiveness

Recent Posts

Smart Home Devices Every Individual Must Have

What is a VPN and Why You May Need One

Mobile Casinos: Then and Now

The best screen recorders in 2021. What will be popular in 2022?

Benefits of Choosing an Unlimited data Plan

Interview Questions

Interview Questions on Red Hat Cluster with Answers

Interview Questions on VMware ESXi with Answers

Interview Questions on Linux Servers with Answers

Linux Interview General Questions with Answers

Interview Questions on Linux Permissions with Answers

Home Privacy Policy Disclaimer Contact Me