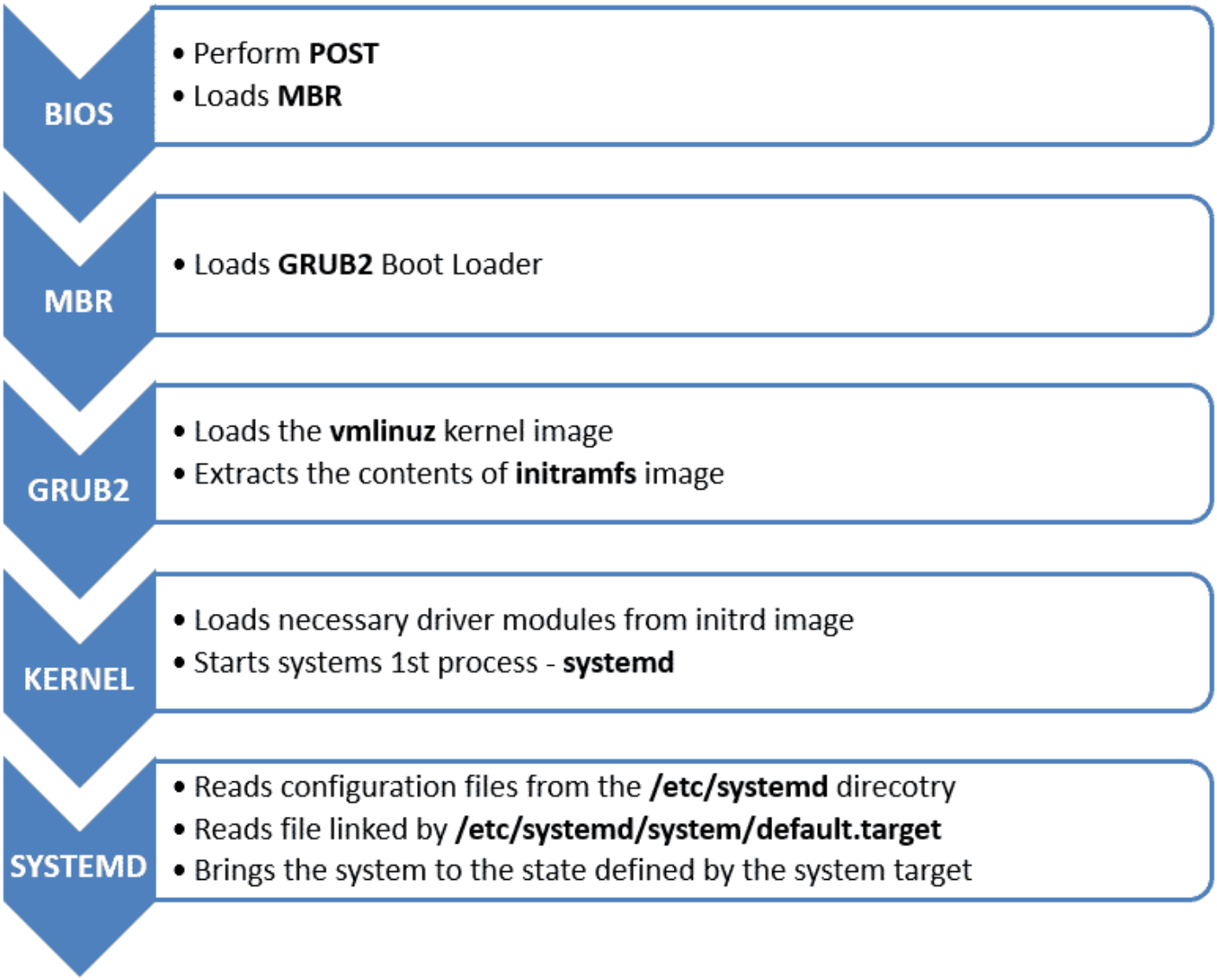
Step By Step Red Hat Enterprise Linux 7 Booting Process copy

Red Hat Enterprise Linux 7 Booting Process

RHEL goes through the boot process when the system is powered up or reset, with the boot process lasting until all enabled services are started and a login prompt appears on the screen.

The following diagram shows the high level stages of a typical Linux boot process:



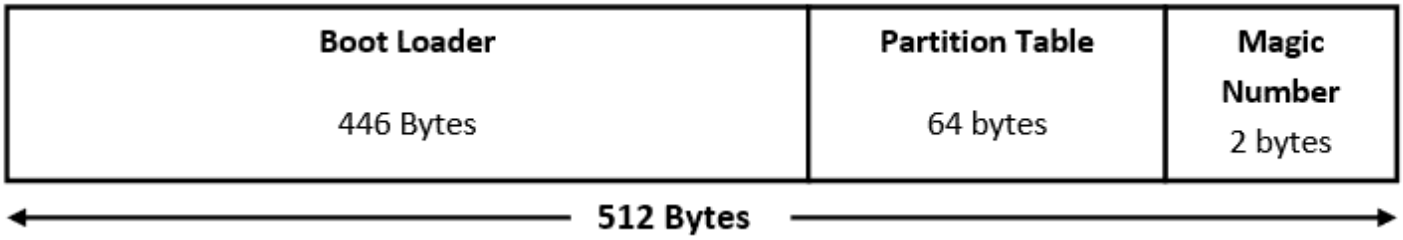
BIOS

- BIOS stands for Basic Input/output System
- Performs some system integrity checks

- Searches, loads, and executes the boot loader program.
- It looks for boot loader in floppy, CD-ROMs, or hard drive. You can press a key (typically F12 of F2, but it depends on your system) during the BIOS startup to change the boot sequence.
- Once the boot loader program is detected and loaded into the memory, BIOS gives the control to it.
- So, in simple terms BIOS loads and executes the **MBR** boot loader.

MBR

- MBR stands for Master Boot Record.
- It is located in the 1st sector of the bootable disk. Typically /dev/hda, or /dev/sda
- MBR is less than **512 bytes** in size. This has three components as shown:
  - Primary boot loader info in 1st 446 bytes
  - Partition table info in next 64 bytes
  - MBR validation check in last 2 bytes.
- It contains information about GRUB2 (GRUB or LILO in old systems).
- So, in simple terms **MBR loads and executes the GRUB2 boot loader**.



GRUB2

- The default bootloader program used on RHEL 7 is **GRUB 2**. GRUB stands for Grand Unified Bootloader. GRUB 2 replaces the older GRUB bootloader also called as legacy GRUB.
- The GRUB 2 configuration file is located at **/boot/grub2/grub.cfg** (Do not edit this file directly).
- GRUB 2 menu-configuration settings are taken from **/etc/default/grub** when generating grub.cfg.
- Sample /etc/default/grub file:

```
[root@yoinsights system]#cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
```

```
GRUB_CMDLINE_LINUX="crashkernel=auto rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
[root@yoinsights system]#
```

- If changes are made to any of these parameters, you need to run **grub2-mkconfig** to re-generate the **/boot/grub2/grub.cfg**

```
# grub2-mkconfig –o /boot/grub2/grub.cfg
```

- GRUB2 searches the compressed kernel image file also called as vmlinuz in the **/boot**
- GRUB2 loads the **vmlinuz** kernel image file into memory and extracts the contents of the **initramfs** image file into a temporary, memory-based file system (tmpfs).
- The initial RAM disk (Initrd) is an initial root file system that is mounted before the real root file system.

About Initramfs – /etc/dracut.conf

- The job of the initial RAM file system is to preload the block device modules, such as for IDE, SCSI, or RAID, so that the root file system, on which those modules normally reside, can then be accessed and mounted.
- The initramfs is bound to the kernel and the kernel mounts this initramfs as part of a two-stage boot process.
- The **Dracut** utility creates **initramfs** whenever a new kernel is installed.
- Use the **lsinitrd** command to view the contents of the image created by dracut:

```
#lsinitrd | less
```

- So, in simple terms **GRUB2 just loads and executes Kernel and initramfs images**.

Kernel

- Linux Kernel is the central core of the OS and it is the first program loaded on the system starts up. While system starting kernel loads all the necessary Kernel Modules and Drives from **initrd.img** to load system first process systemd in Linux 7.
- The kernel starts the systemd process with a process ID of **1** (PID 1) as shown below:

```
#ps -ef |grep -i systemd
root    1    0  0 02:10 ?   00:00:02 /usr/lib/systemd/systemd --switched-root
```

systemd

- **Systemd** process is the first process ID (PID 1) to run on Linux 7 systems, it initializes the system and launches all the services that were once started by the traditional init

system and launches all the services that were once started by the traditional init (/etc/init.d) process. Systemd process reads the configuration file of **/etc/systemd/system/default.target**, then its load the OS in targeted **target**.

- This tells systemd to start everything in the /usr/lib/systemd/system/basic.target before starting the other multi-user services.
- systemd brings the system to the state defined by the system target, performing system initialization tasks such as:
- Setting the host name
- Initializing the network
- Initializing **SELinux** based on its configuration
- Printing a welcome banner
- Initializing the system hardware based on kernel boot arguments
- Mounting the file systems, including virtual file systems such as the /proc file system
- Cleaning up directories in /var
- Starting swapping
- systemd uses 'targets' instead of runlevels. By default, there are two main targets:
  - multi-user.target: analogous to runlevel 3
  - target: analogous to runlevel 5

Note: Refer Man pages for more details: **info grub2**

Boot, Reboot and Shutdown

To power off or reboot a running system from the command line, we can use the **systemctl** command.

- **systemctl poweroff** will stop all running services, unmount all file systems (or remount them read-only when they cannot be unmounted), and then power down the system.
- **systemctl reboot** will stop all running services, unmount all file systems, and then reboot the system.

Note: systemctl halt and **halt** are also available to stop the system, but unlike their poweroff equivalents, these commands do not power off the system; they bring a system down to a point where it is safe to manually power it off.

Managing GRUB

The /etc/default/grub File

The **/etc/default/grub** configuration file defines directives that govern how GRUB behaves at boot time. Values defined in this file are used to regenerate the **/boot/grub2/grub.cfg** file, which controls the behavior of GRUB at boot time. Any changes made to the grub file will only take effect after the **grub2-mkconfig** utility has been executed.

```
[root@yoinsights system]#cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
[root@yoinsights system]#
```

Directive	Description
GRUB_TIMEOUT	Sets the wait time, in seconds, before booting off the default kernel. Default value is 5.
GRUB_DISTRIBUTOR	Defines the name of the Linux distribution.
GRUB_DEFAULT	Boots the selected option from the previous system boot.
GRUB_DISABLE_SUBMENU	Enables/disables the appearance of GRUB submenu.
GRUB_TERMINAL_OUTPUT	Sets the default terminal.
GRUB_CMDLINE_LINUX	Specifies the command line options to pass to the kernel at boot time.
GRUB_DISABLE_RECOVERY	Disables showing system recovery entries in the GRUB menu

The /boot/grub2/grub.cfg File

This is the main configuration file that controls the behavior of GRUB at boot time. This file is located in the **/boot/grub2** directory on BIOS-based systems and in **the /boot/efi/EFI/redhat** directory on UEFI-based systems. This file can be regenerated manually with the **grub2-mkconfig** utility, or it is automatically regenerated when a new kernel is installed. During this process, any manual changes made to this file are lost.

Here is how you would run this utility to reproduce the grub.cfg file on BIOS and UEFI systems, respectively:

```
# grub2-mkconfig –o /boot/grub2/grub.cfg
# grub2-mkconfig –o /boot/efi/EFI/redhat/grub.cfg
```

When this utility runs, it uses the settings defined in the **/etc/default/grub** file and in the

helper scripts located in the **/etc/grub.d/** directory to regenerate this file for kernels located in the **/boot** directory. Here is the list of default helper scripts located in the /etc/grub.d/ directory:

```
[root@yoinsights ~]#ll /etc/grub.d
total 68
-rwxr-xr-x. 1 root root  8702 Mar 26  2015 00_header
-rwxr-xr-x. 1 root root   992 Oct 17  2014 00_tuned
-rwxr-xr-x. 1 root root 10114 Mar 26  2015 10_linux
-rwxr-xr-x. 1 root root 10275 Mar 26  2015 20_linux_xen
-rwxr-xr-x. 1 root root  2559 Mar 26  2015 20_ppc_terminfo
-rwxr-xr-x. 1 root root 11169 Mar 26  2015 30_os-prober
-rwxr-xr-x. 1 root root   214 Mar 26  2015 40_custom
-rwxr-xr-x. 1 root root   216 Mar 26  2015 41_custom
-rw-r--r--. 1 root root   483 Mar 26  2015 README
[root@yoinsights ~]#
```

The first script, **00_header**, sets the GRUB environment; the **10_linux** script searches for all installed kernels on the same disk partition; the **30_os-prober** searches for the presence of other operating systems; and the **40_custom** and **41_custom** scripts are for us to add any customization to the new GRUB configuration file.

The **grub.cfg** file contains menuentry blocks for each installed kernel. Each block begins with a **title** and includes the names of the **kernel** and **RAM disk image** files, their **location** with respect to /**boot**, and several options and modules to be loaded. These menu entry titles are displayed at the time of system boot and you can choose one of them to boot.

```
menuentry 'CentOS Linux 7 (Core), with Linux 3.10.0-229.el7.x86_64' --class
rhel fedora --class gnu-linux --class gnu --class os --unrestricted
$menuentry_id_option 'gnulinux-3.10.0-229.el7.x86_64-advanced-69a4df5e-3dfc-
4c03-a4ab-18a99e892aa8' {
        load_video
        set gfxpayload=keep
        insmod gzio
        insmod part_msdos
        insmod xfs
        set root='hd0,msdos1'
        if [ x$feature_platform_search_hint = xy ]; then
          search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --
hint-efi=hd0,msdos1 --hint-baremeta
l=ahci0,msdos1 --hint='hd0,msdos1'  a8045228-dcf6-4e2a-bddc-65782638aeaf
```

```
        else
          search --no-floppy --fs-uuid --set=root a8045228-dcf6-4e2a-bddc-
65782638aeaf
        fi
        linux16 /vmlinuz-3.10.0-229.el7.x86_64 root=UUID=69a4df5e-3dfc-4c03-
a4ab-18a99e892aa8 ro crashkernel=auto rhgb quiet LANG=en_US.UTF-8
        initrd16 /initramfs-3.10.0-229.el7.x86_64.img
}
```

If a new kernel is added to the system, existing kernel entries will remain in this file and can be chosen in the GRUB menu at startup to boot.

Important Directives:

- To troubleshoot a broken **grub2** configuration, an administrator will need to understand the syntax of **/boot/grub2/grub.cfg** Actual bootable entries are encoded inside menuentry blocks. In these blocks, **linux16** and **initrd16** lines point to the kernel to be loaded from disk and the **initramfs** to be loaded.
- The **set root** lines inside those blocks do not point to the root file system for the RHEL 7 system, but instead point to the file system from which grub2 should load the kernel and initramfs files. The s

```
#systemctl list-units - -type=target - -all
```

- yntax is (**harddrive**, **partition**) where **hd0** is the first hard drive in the system, **hd1** is the second etc… the partitions are indicated as **msdos1** for the 1st MBR partition or **gpt1** for the 1st GPT partition on that drive.

Selecting a systemd target (runlevel in older version)

A systemd target is a set of systemd units that should be started to reach a desired state. The most important of these targets are listed in the following table:

Target	Purpose
graphical.target	System supports multiple users, graphical and text-based logins.
multi-user.target	System supports multiple users, text-based logins only.
rescue.target	**sulogin** prompt, basic system initialization completed.
emergency.target	**sulogin** prompt, **initramfs** pivot complete and system root mounted on **/** read-only.

- To view all available targets

- To view all targets installed on disk

```
#systemctl list-unit-files - -type=target - -all
```

- Selecting a target at runtime

On a running system, we can choose to switch to a different target using the **systemctl isolate** command as shown.

```
#systemctl isolate multi-user.target
```

Isolating a target will stop services not required by that target and start any required services that have not yet been started.

Note: Not all targets can be isolated. Only targets that have **AllowIsolate=yes** set in their unit files can be isolated.