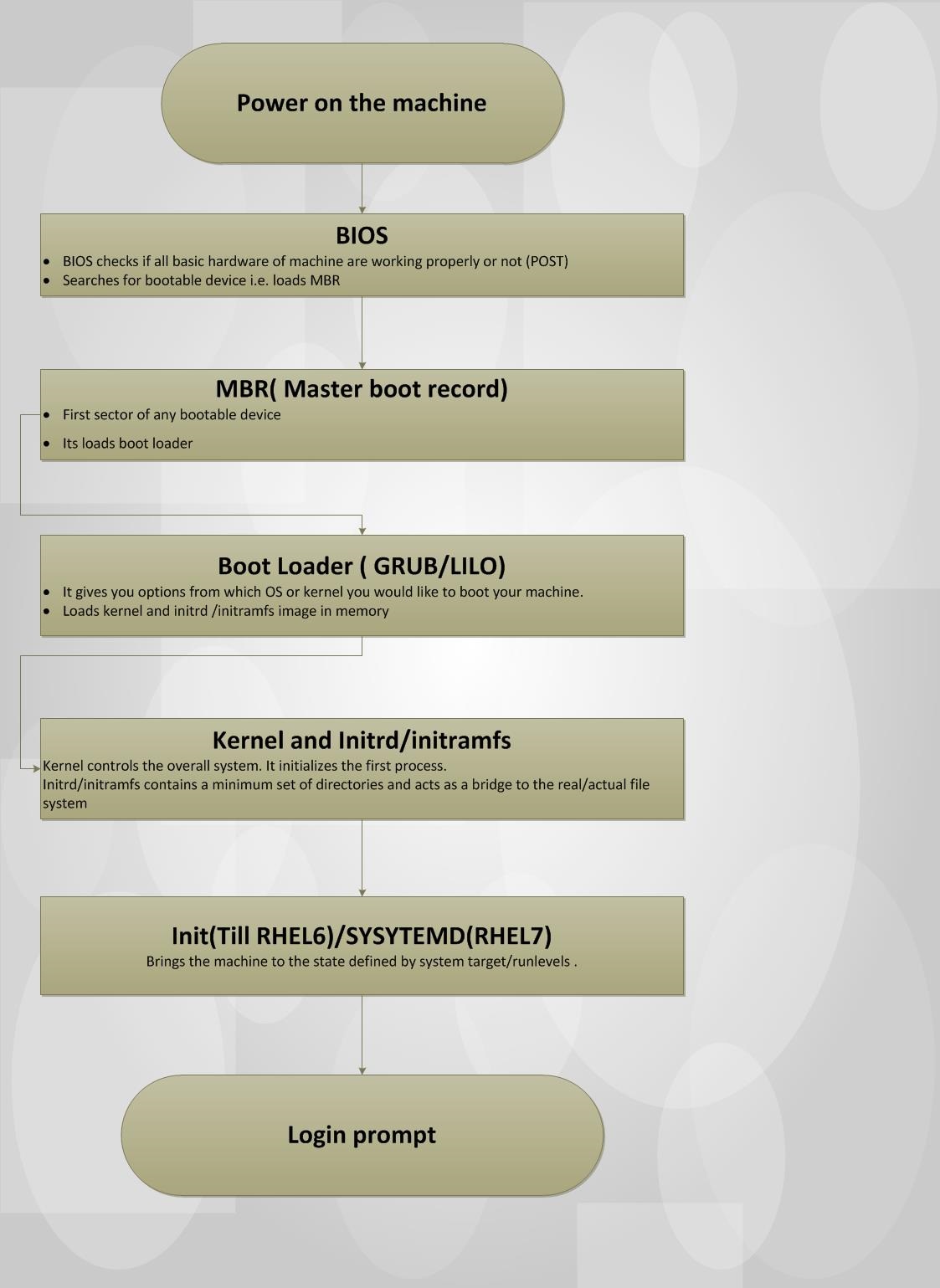
**Step by Step Linux (RHEL 6 / 7) boot process:**

* Boot process consists the set of processes from power on the pc to login prompt comes.
* Press the power button on your system, and after few moments you see the Linux login prompt.
* The following high-level stages of a typical Linux boot process.

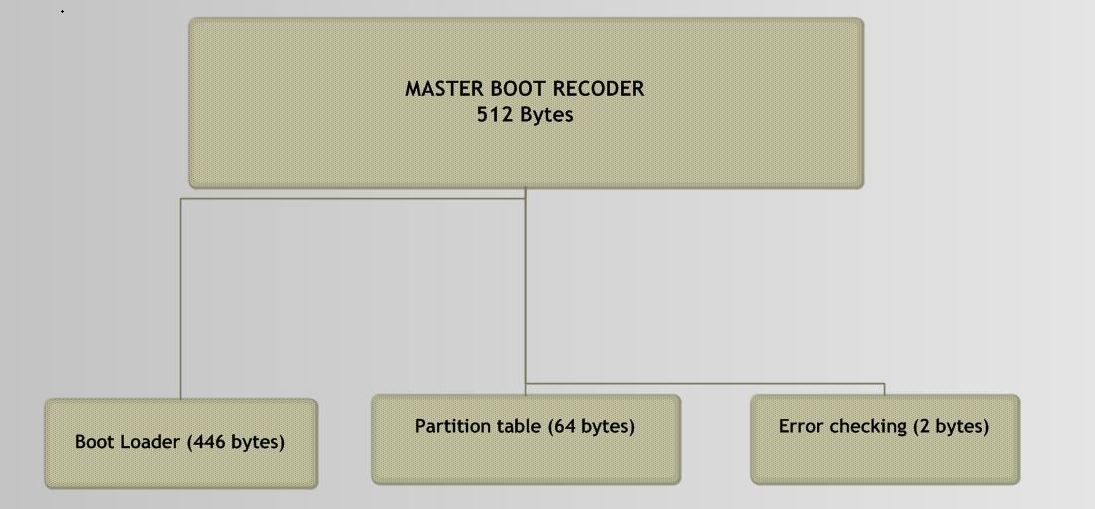
[](http://www.linuxbuzz.com/wp-content/uploads/2018/01/Linux-boot-process-for-RHEL-6-7-.jpg)

**BIOS:**

* BIOS stands for Basic Input/output System.
* When we power on BIOS performs a POST (Power-on-self-Test) for all of the hardware components in the system to make sure everything is working properly.
* BIOS searches for MBR. So, in simple terms BIOS loads and executes the MBR boot loader.
* hardware that includes UEFI, as most of the machine now comes with UEFI. UEFI support **GPT** partition instead of 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 1) primary boot loader info in 1st 446 bytes 2) partition MBR table info in next 64 bytes 3) mbr validation check in last 2 bytes.
* It contains information about GRUB2 (or GRUB in old systems).

[](http://www.linuxbuzz.com/wp-content/uploads/2018/01/MBR-Linux-Boot-Process.jpg)

**Boot loader (GRUB2):**

* GRUB stands for Grand Unified Boot Loader.
* In this stage, if you have multiple kernel images installed on your system, you can choose which one to be executed.
* GRUB displays a splash screen, waits for few seconds, if you don’t enter anything, it loads the default kernel image as specified in the grub configuration file.

[](http://www.linuxbuzz.com/wp-content/uploads/2018/01/Bootloader-Linux-Boot-Process.jpg)

In the above image, we can see we are getting 2 options from which we can boot machine.

* Its main job is to load kernel and initrd /initramfs image in memory.
* Once boot loader loads kernel in RAM it passes control to Kernel.

**NOTE: In RHEL7**

1. The default boot loader is GRUB 2 (Grand Unified Bootloader).
2. The configuration file is  located at  /boot/grub2/grub.cfg
3. Menu-configuration setting is located at /etc/default/grub.

**Kernel:**

* Initially, the kernel is loaded in read-only mode.
* Kernel initializes the first process that is init (Till RHEL6) /Systemd (from RHEL 7).
* The kernel manages memory, input-output device. In short, it controls the overall system.
* Kernel is compiled with different drivers, but they are not able to load actual file system because some     additional drivers are required for it. For this purpose, intird/ iniramfs is loaded in RAM, so that it can install other drivers which are required to load actual file system.
* Initramfs/initrd gets decompressed and then it first loads temporary file system.

**Initrd /Initramfs:**

* It loads temporary root file system into memory.
* It contains a minimum set of directories and acts as a bridge to the real/actual file system.
* Is detects what all device drivers are needed to load the actual file system and it loads them from a temporary   file system. After that, other partition like LVM, RAID etc is mounted then initrd is unmounted.
* We can see kernel and initramfs file in /boot directory. Staring with vmlinuz denotes kernel.
* Ls /boot

**Difference between Initramfs and initrd:**

* Successor of initrd (initial ram disk) is Initramfs (initial ram disk file system). Feature made up from a cpio (copy input and output) archive of files that enables an initial root filesystem and init program to reside in kernel memory cache, rather than on a ramdisk, as with initrd filesystems.
* Initramfs is used by 2.6 kernels whereas initrd was used by older 2.4 (and earlier) kernels.

Once the real file system is mounted, then kernel runs the first process **i.e init(till RHEL 6) / Systemd from (RHEL 7).**

**Init**

* It is the first process that runs in the system with process id of 1.
* Can check with ps –ef  | grep init.
* Also known as the parent process, responsible for starting all the other process.
* Init is responsible for bringing the computer into the normal running state after power on, also gracefully shutting down services prior to shut down.
* Init starts services to the corresponding runlevel on which our machine is booted.
* Now you might be thinking what are run levels? Let’s us check.

**Runlevel** is a state of a machine that defines how a machine should be login, what services and scripts should run when a machine starts.  Runlevel are defined from 0 – 6.

1. Runlevel 0 – shut down the system
2. Runlevel 1 – single mode
3. Runlevel 2 – multiuser mode without networking
4. Runlevel 3 – multiuser with text login screen
5. Runlevel 4- customized runlevel (not in use)
6. Runlevel 5 – runlevel 3 with graphical login
7. Runlevel 6 – Reboots the system

* We can set in which runlevel we want to run our operating system by defining it on /etc/inittab file.
* Typically, you would set the default run level to either 3 or 5.
* But from RHEL 7   init and runlevels are replaced by systemd and targets respectively.

**Systemd**

* Same as init it is the first service started by the kernel, with process id 1.
* Daemons are the programs that run in the background performing various tasks.
* Services usually refer to one or more daemons.
* Configuration file: **/etc/systemd**.

**Init had several limitations that were taken care by sytemd.**

Few of them are:

1. In init, one script runs at a time, while in systemd multiple scripts run at a time parallel. Thus, increasing the performance.
2. In init, an administrator has to take care of what scripts should run first. Init process does not know if the scripts defined in particular runlevel are dependent on another script, while systemd takes care of dependency.
3. Systemd kills daemon properly before the machine shuts down.
4. inittab is no longer used when using systemd.
5. systemd uses ‘targets’ instead of runlevels. By default, there are two main targets:

a)    multi-user.target: analogous to runlevel 3

b)   graphical.target: analogous to runlevel 5

6.  Ctrl-Alt-Delete is handled by /usr/lib/systemd/system/ctrl-alt-del.target

**To view current default target, run**:

#systemctl get-default

**To get default target, run:**

#systemctl set-default TARGET.target

Managed services of the system are controlled by an application known as **systemctl.**

Like **starting/stoping services, enable/disable services**, checking the status of services etc.

systemctl start ssh (starting the service of ssh)

Systemctl status ssh (checking their service status if it is running, stopped, enabled, disabled).