BOOTING PROCESS

What is booting process?

Did you ever wonder when you click on power-on button in your system, what will happen? There is some magic(logic) going inside for 1-2 minutes and then you will see the login screen. That magic/logic is called as booting process.

*"**Booting** is a **process** or set of operations that loads and hence starts the operating system, starting from the point when user switches on the power button"*

Why it is needed?

It is very important to know about Linux booting process of RHEL/CentOS 7 to troubleshoot the booting problem and understanding the Linux OS functionality.

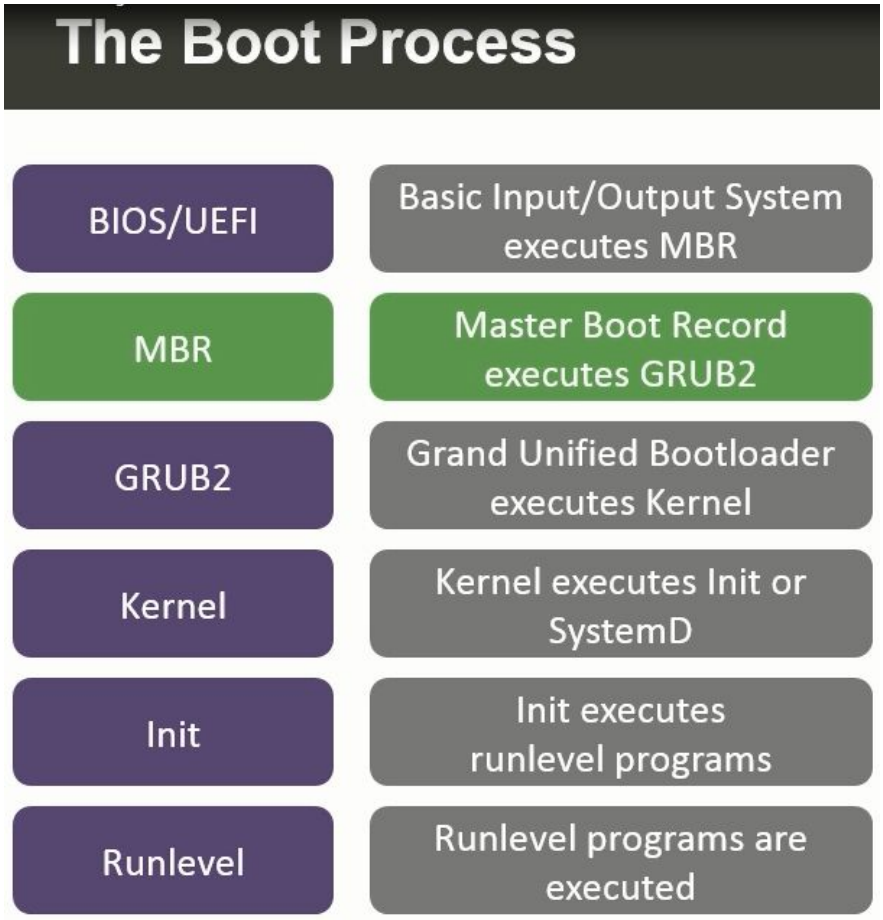
Having a solid understanding of boot process is crucial

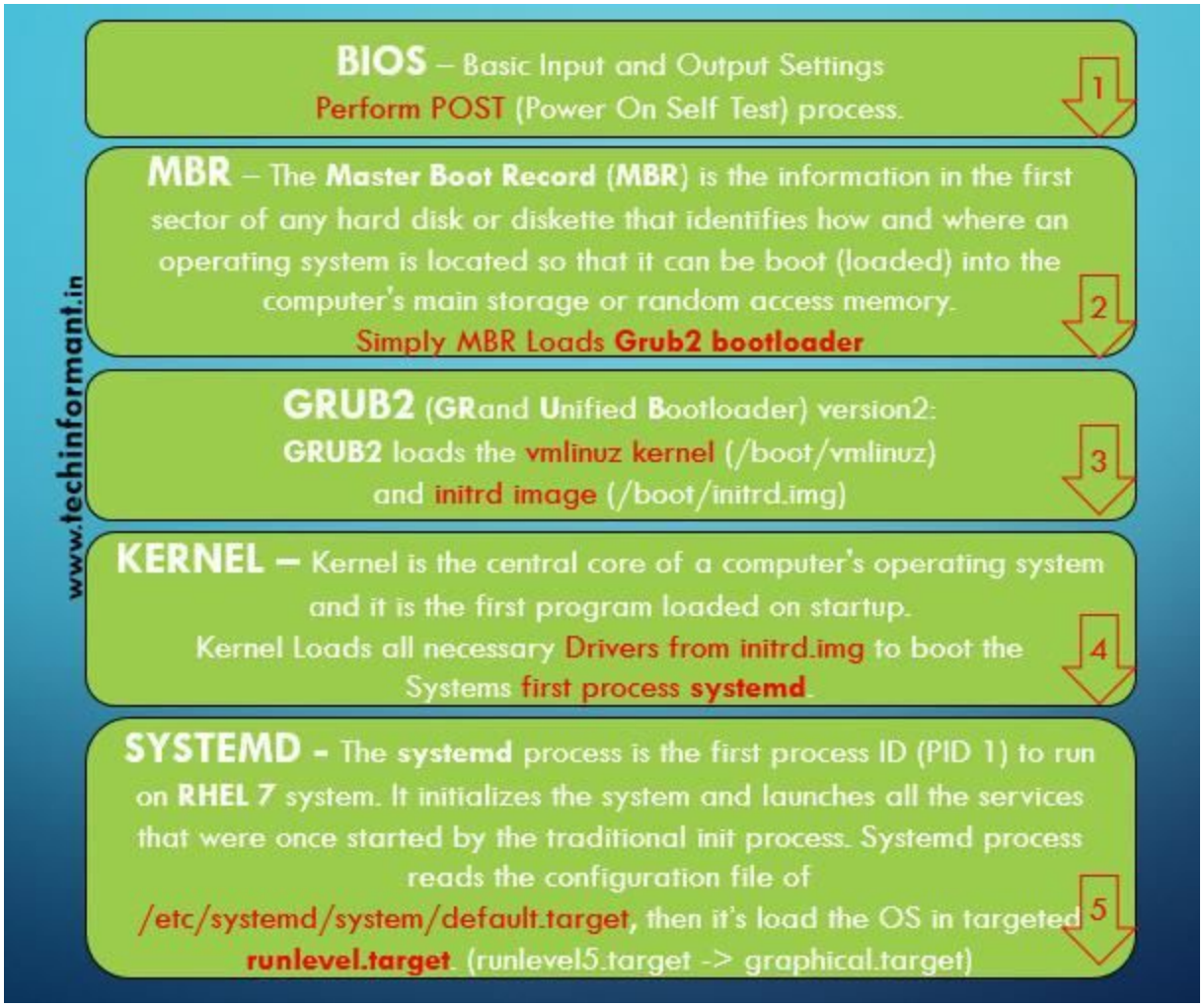
1) If you want to customize the linux startup env for eg: by having multiple boot options for different OS
2) Troubleshooting startup issues

How?

First Click on power-on button

Below are the 6-high level phases of a typical linux boot process





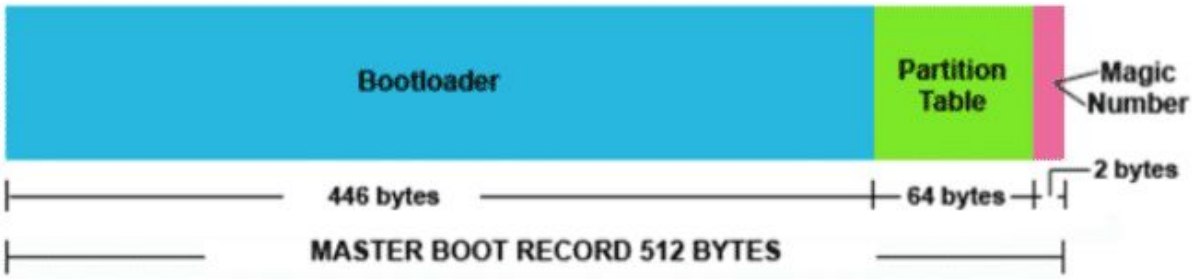
1. BIOS

- BIOS stands for Basic Input Output system, UEFI stands for **Unified Extensible Firmware Interface**
- Basic Input/Output System is a firmware interface (software stored on a small memory chip i.e ROM on the motherboard) that controls not only booting process but also provides all the control of low-level interface to attached peripheral devices
- When you power on your system, it will read all the devices settings and executes the POST (Power ON Self-Test) process to detect,test and initialize the system Hardware components

- After successful POST process, it looks for boot loader in floppy, cd-rom or hard drive. You can press a key (typically F12 or F2, depends on your system) during BIOS startup phase to change the boot sequence
- Once the boot loader program is detected then it will load into memory, So BIOS is giving the control to it.
- **IN SIMPLE TERMS, BIOS LOADS AND EXECUTE THE MBR BOOT LOADER**

2. Master boot record (MBR)

- MBR stands for Master Boot Record. This MBR is read into memory by the BIOS
- 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 (machine code instructions)
    - 2) partition table info in next 64 bytes
    - 3) mbr validation check in last 2 bytes. It is also called as magic number which is used for error detection



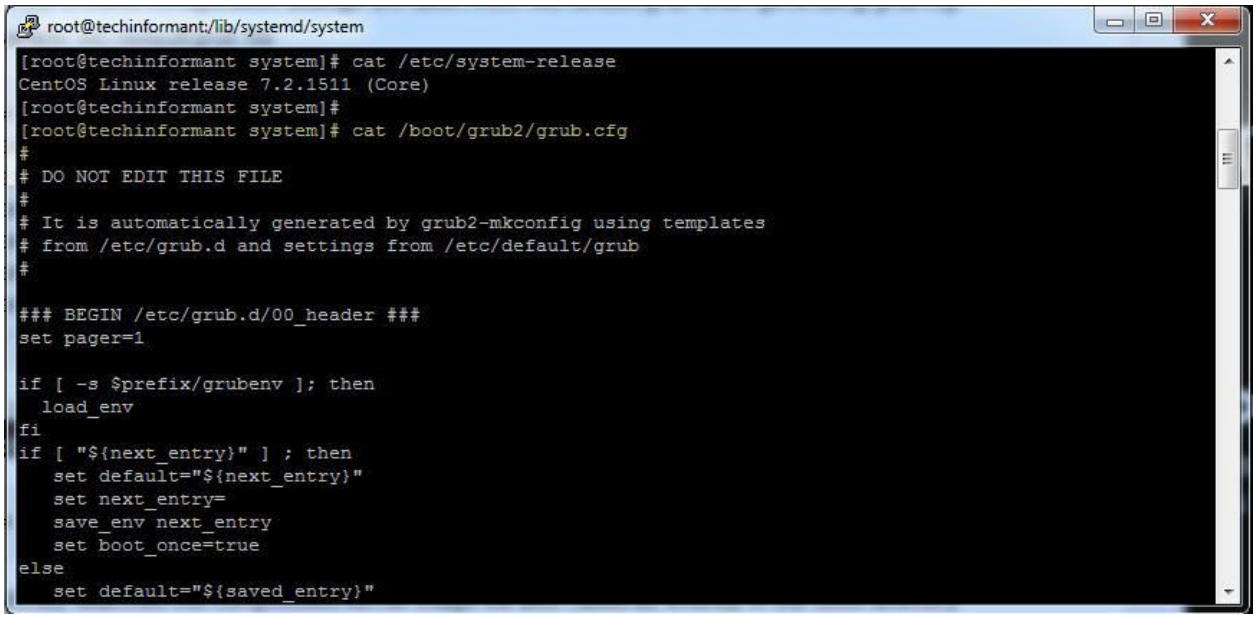
- **It contains information about GRUB2 (GRUB in old systems)**
- MBR discovers the bootable device and loads the GRUB2 boot loader into memory and transfers control over to it.
- **In simple Terms, MBR (Master Boot Record) loads and executes the GRUB2 bootloader.**

3. GRUB2

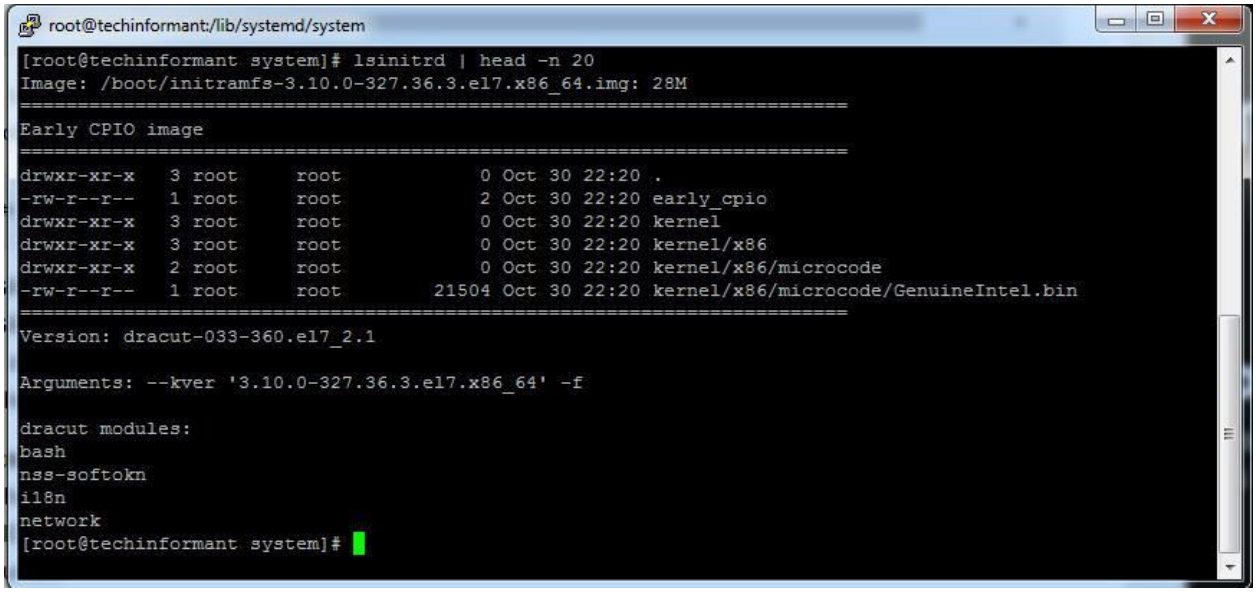
- GRUB2 stands for Grand Unified Bootloader version2

- If you have multiple kernel images installed on your system, you can choose which one to be executed.
- GRUB displays a splash screen asking for confirmation of which system to load (allows for multiple OS installations on a single computer, if you don't enter anything, it loads the default kernel image as specified in the grub configuration file.
- It contains info on the kernel and any Initial RAM Disk (initrd) required to complete the booting process
- GRUB2 is the default bootloader program in all latest version of like Red Hat/CentOS 7 and also Ubuntu from version 9.10. It has been replaced by GRUB bootloader also known as GRUB legacy.
- GRUB2 configuration file located in **/boot/grub2/grub.cfg** and It is automatically generated by grub2-mkconfig using templates from /etc/grub.d and settings from **/etc/default/grub**. Do not recommend to edit the GRUB2 configuration file.

Following is the example



The boot loader (GRUB2 for RHEL 7) starts the RHEL 7 kernel and initial RAM disk(initrd). GRUB 2 is installed in the boot sector of your server's hard drive and is configured to load a Linux kernel and the initramfs and the initrd is an initial root file system that will mount prior to the real root file system on Linux system.



In simple terms GRUB just loads and executes Kernel and initrd images

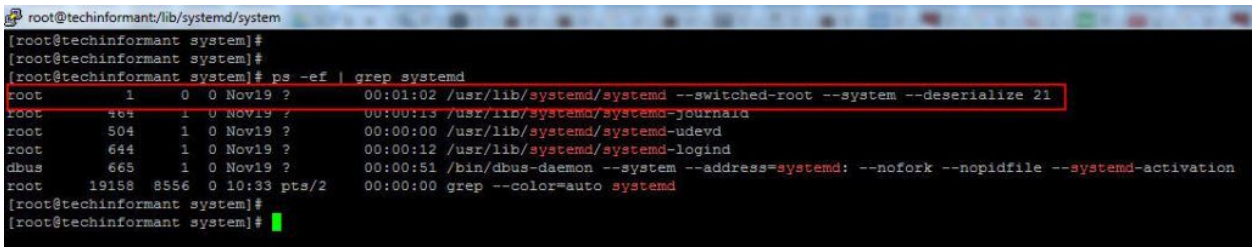
In case,if we change the boot parameters in /etc/default/grub then we need to create a new /boot/grub2/grub.cfg file using

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

4. 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 Drivers from initrd.img to load system first process systemd in Linux 7.

Below command will help you find the systemd process id(PID)



Basically kernel executes initialization scripts (Older system it was init process and Newer RHEL 7 system have SystemD process)

Kernel then uses initial RAMdisk (initrd) as a temporary File system until the actual physical file system is mounted and available.

The kernel either contains the hardware drivers or loads the modules for the hardware from the initrd or from the mounted partitions.

5. init

- Init process is used in older linux environment but in newer environment such as RHEL 7 using SystemD instead of init
- Systemd (/etc/systemd/) is the replacement of init (/sbin/init) process
- The very first process (PID 1) for a Linux system is used by init
- Running services are then managed by the init process
- Most init command (/sbin/init) will work seamlessly with SystemD

6. Runlevel

- Linux systems supports something called runlevel
- Run levels are used to specify services that are started or stopped based on a numeric value from 0 to 6

Eg:

- Runlevel 0 is almost always used to run shutdown scripts
- Runlevel 5 is usually used for a graphical boot (eg. GNOME)
- Runlevel 3 is usually used for a non-graphical services with all network services on.
- Runlevel 6 is usually used for rebooting
  SystemD also has runlevels but they are referred to as "**targets**"
- runlevel0.target
- runlevel5.target
-

CONFIGURING BOOT LOADER

In linux, boot loader is what kicks in when we power-on the machine prior to the OS is being loaded. It gives us the menu of the options to choose from, even booting other operating system.
In Modern Linux OS, we are using grub version 2 ie grub2 is used.

Other than configuring the menu items on the screen we can also configure the kernel parameters while booting.

Example1: Please change the menu waiting time from 5 secs to 20 secs

Do not change the direct boot file /boot/grub2/grub.cfg

Instead change the configuration file under

#vi /etc/default/grub
GRUB_TIMEOUT=5
Change this above parameter to
GRUB_TIMEOUT=20

Save and Exit

So after making changes to our boot loader config file we need to create a new grub.cfg file using the below command
#grub2-mkconfig -o /boot/grub2/grub.cfg

This will create a new grub config file

Then reboot to see , will it effect or not
#init 6
Verify :
You can see that the menu wait for 20 secs instead of 5 sec

Example 2: Please disable the IPv6 kernel parameters while booting

```
[root@rhel3 ~]# ifconfig
eno16777736: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.244  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::20c:29ff:fe3c:a05d  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:3c:a0:5d  txqueuelen 1000  (Ethernet)
        RX packets 2203  bytes 147986 (144.5 KiB)
        RX errors 0  dropped 1  overruns 0  frame 0
        TX packets 162  bytes 14944 (14.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 0  (Local Loopback)
        RX packets 4  bytes 420 (420.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 420 (420.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@rhel3 ~]#
```

Do not change the direct boot file /boot/grub2/grub.cfg

Instead change the configuration file under
#vi /etc/default/grub

```
GRUB_TIMEOUT=20
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root rhgb quiet ipv6.disable=1 "
GRUB_DISABLE_RECOVERY="true"
```

Add this parameter **ipv6.disable=1**
Save and Exit

Now when we boot our system this kernel parameters will be passed to the linux kernel and the kernel will behave in the manner based on the parameters we passed.
So after making changes to our boot loader config file we need to create a new grub.cfg file using the below command
#grub2-mkconfig -o /boot/grub2/grub.cfg

This will create a new grub config file

Then reboot to see , will it effect or not
#init 6

Verfiy :

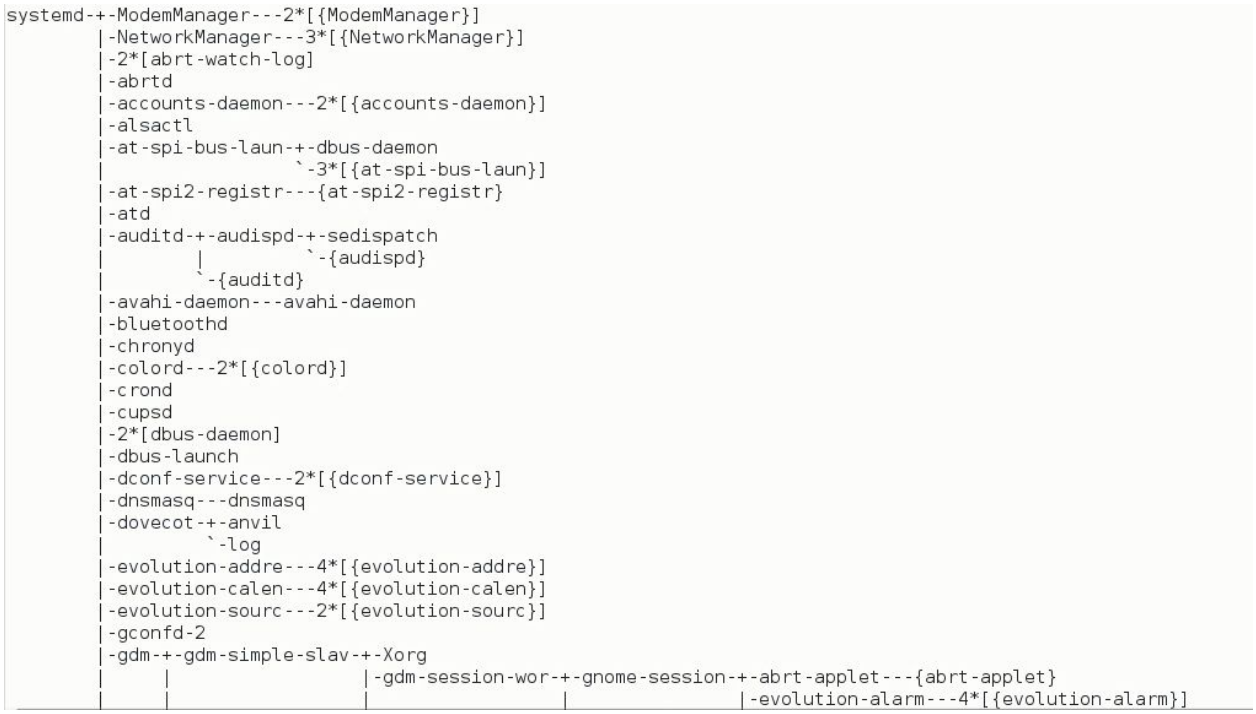
```
[root@rhel3 ~]# ifconfig
eno16777736: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.244  netmask 255.255.255.0  broadcast 192.168.1.255
        ether 00:0c:29:3c:a0:5d  txqueuelen 1000  (Ethernet)
        RX packets 219  bytes 16135 (15.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 44  bytes 4816 (4.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

You can also do it by other way also i.e., changing the kernel parameters in /etc/sysctl.conf file
https://www.techrepublic.com/article/how-to-disable-ipv6-on-linux/

SETTING RUNLEVEL

In Linux, runlevels allow us to determine which daemons will automatically start or shutdown. They also determine some capabilities, for example
Runlevel 3 allows multiple users to be logged on to the system with network support whereas runlevel 5 allows the same thing but in addition to it start the graphical x windows environment

Here is the command which list the processes in hierarchy format
#pstree |more



In RHEL 7, systemd is the first process under which all processes will spawn.
In previous version, that was init daemon, we don't have init daemon running by default here now.

Now check the /etc/inittab file

In previous version of RHEL, we would set the default runlevel in this file when the system was booting Now because RHEL 7 is using systemd by default, this file is no longer being used.
However , from the command line, I can still use the command as runlevel
runlevel

Runlevel command will display the previous and current runlevel

```
[root@rhel1 ~]# vi /etc/inittab
[root@rhel1 ~]# runlevel
3 5
[root@rhel1 ~]#
```

Here current runlevel is 5 and previous runlevel was 3

Using the init command line, even though its not managing processes, it still exists on hard drive and i can use it to switch to different runlevels.

#init 3
This will switch the current runlevel of 5 to runlevel 3

Login and check the runlevel and change it back to runlevel 5

To reboot the system

#init 6

Or You can also use the below command to reboot
#shutdown -r now

If you want to reboot the server and want to give some message to login users and time to save their data then run below command
#shutdown -r +5 “software upgrades”

To shutdown the server
#shutdown -h now or #poweroff

-h means halt here

CHANGING BOOT TARGET

In RHEL 7, systemd is now the process that spawns all the processes when the OS boots.
In previous version, init daemon did all this jobs. So init daemon runs runlevel whereas runlevel 3 means all services with network support and runlevel5 means same + graphical support
Systemd doesn't use those runlevels instead it uses called as "target"
We can get the list of available target by using the below command

#systemctl list-units -t target

```
[root@rhel1 ~]# systemctl list-units -t target
UNIT                   LOAD   ACTIVE SUB    DESCRIPTION
basic.target           loaded active active Basic System
cryptsetup.target      loaded active active Encrypted Volumes
getty.target           loaded active active Login Prompts
graphical.target       loaded active active Graphical Interface
local-fs-pre.target    loaded active active Local File Systems (Pre)
local-fs.target        loaded active active Local File Systems
multi-user.target      loaded active active Multi-User System
network.target         loaded active active Network
nss-lookup.target      loaded active active Host and Network Name Lookups
nss-user-lookup.target loaded active active User and Group Name Lookups
paths.target           loaded active active Paths
remote-fs-pre.target   loaded active active Remote File Systems (Pre)
remote-fs.target       loaded active active Remote File Systems
slices.target          loaded active active Slices
sockets.target         loaded active active Sockets
swap.target            loaded active active Swap
sysinit.target         loaded active active System Initialization
timers.target          loaded active active Timers

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

18 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
[root@rhel1 ~]#
```

Here
Multi-user.target means runlevel 3 in previous
Graphical.target means runlevel 5 in previous

We can determine, what is the default target is on this system.

#systemctl get-default

```
[root@rhel1 ~]# systemctl get-default
graphical.target
[root@rhel1 ~]#
```

This means that when this system is booted,it will automatically comes into graphical mode.

If you want to change it multi-user target then

#systemctl set-default multi-user.target

```
[root@rhel1 ~]# systemctl set-default multi-user.target
rm '/etc/systemd/system/default.target'
ln -s '/usr/lib/systemd/system/multi-user.target' '/etc/systemd/system/default.target'
[root@rhel1 ~]#
```

It will remove the old default target and add the new link to the appropriate target

Verify:

```
[root@rhel1 ~]# systemctl get-default
graphical.target
[root@rhel1 ~]# systemctl set-default multi-user.target
rm '/etc/systemd/system/default.target'
ln -s '/usr/lib/systemd/system/multi-user.target' '/etc/systemd/system/default.target'
[root@rhel1 ~]# systemctl get-default
multi-user.target
[root@rhel1 ~]#
```

Now by default system will boot into multi-user mode.

How to go to single user mode(also known as Maintenance mode or emergency mode)

With the single user mode , we can perform maintenance on the system. It did not ask for username and password at the time of login. So this can also be used for reset the root password if anyone forgets it.

Booted up the linux and stop at the menu and select the entry
And press “e”
Now go to line that begins linux16 /vmlinux and at the middle of line write in place of **ro rw init=/sysroot/bin/sh**

Then press Ctrl-x

After a moment, it booted into a small shell without password
It does not know many of the commands in this maintenance mode

#ls /
(it is not same root File System as our original one)

To get to the original root file system in maintenance mode run the below command

#chroot /sysroot
#ls /
Now it shows different one (the real root fs)
#whoami
root
All commands we can run now

To reset the root password
#passwd
Type
Type

References:
https://www.linuxtechi.com/boot-rhel-7-centos-7-server-single-user-mode/
https://access.redhat.com/solutions/918283

TROUBLESHOOTING BOOT FAILURE ISSUES

First thing we can go to single user mode

#ls /
This is not the entire root FS
To get tothe enitre root FS
#chroot /sysroot
#ls /
Access all the item we have on root FS

Check /etc/fstab , if any pb with FS
Check /etc/default/grub , if any pb with boot parameters

TROUBLESHOOTING BOOT FAILURES

First understand the sequence of boot process at high level.
When we power on the server, it did power on self test (POST) task is completed by initializing the hardware and if it succeeds then boot loader gets loaded from the MBR on the Hard Disk. If that succeeds, we normally sees the GRUB boot loader menu which shows us the boot options to boot into various operating systems kernel , or rescue mode or so on. Normally there is a countdown timer in this screen after which it will execute default boot items. We can interrupt that counter using an UP and DOWN arrow keys.
We will first highlight the first entry of kernel and then press “e” to edit that entry
Trying to boot into single user mode to troubleshooting linux systems issues for fixing problems with configuration files or reset a root password or so on.

Go down to the entry which starts with linux16

```
        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 --hin\
t-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1'  f7b5b986-d\
22c-47a2-ad18-1836d5e786c8
        else
          search --no-floppy --fs-uuid --set=root f7b5b986-d22c-47a2-ad18-1836\
d5e786c8
        fi
        linux16 /vmlinuz-3.10.0-229.el7.x86_64 root=/dev/mapper/rhel-root ro r\
d.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root rhgb quiet LANG=en_US.\
UTF-8
        initrd16 /initramfs-3.10.0-229.el7.x86_64.img                       ↓

      Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
      discard edits and return to the menu. Pressing Tab lists
      possible completions.
```

Replace ro with rw init=/sysroot/bin/sh

Then press Ctrl+x

```
         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 --hin\
 t-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1'  f7b5b986-d\
 22c-47a2-ad18-1836d5e786c8
         else
           search --no-floppy --fs-uuid --set=root f7b5b986-d22c-47a2-ad18-1836\
 d5e786c8
         fi
         linux16 /vmlinuz-3.10.0-229.el7.x86_64 root=/dev/mapper/rhel-root rw i\
 nit=/sysroot/bin/sh rd.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root r\
 hgb quiet LANG=en_US.UTF-8
         initrd16 /initramfs-3.10.0-229.el7.x86_64.img                      ↓

       Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
       discard edits and return to the menu. Pressing Tab lists
       possible completions.
```

After a moment, you will be presented with a command prompt without entering any username and password.

```
[    6.884070] sd 32:0:0:0: [sda] Assuming drive cache: write through
[    6.885506] sd 32:0:3:0: [sdd] Assuming drive cache: write through
[    6.885550] sd 32:0:1:0: [sdb] Assuming drive cache: write through
[    6.885622] sd 32:0:2:0: [sdc] Assuming drive cache: write through

Generating "/run/initramfs/rdsosreport.txt"


Entering emergency mode. Exit the shell to continue.
Type "journalctl" to view system logs.
You might want to save "/run/initramfs/rdsosreport.txt" to a USB stick or /boot
after mounting them and attach it to a bug report.

:/#
```

If you type ls then you will see some folders but this is not the real ones as your original system had.

To get to the entire root file system, I can type the below command
#chroot /sysroot/

Now if you type ls then we can see the original system folders.

```
:/# ls /
bin  dracut-state.sh  etc   kernel  lib64  root  sbin      sys      tmp  var
dev  early_cpio       init  lib     proc   run   shutdown  sysroot  usr
:/# chroot /sysroot
:/# ls /
bin      dump          lib    net             Remote_website  sbin      swapfile   usr
boot     dumpfile_xfs  lib64  opt             removable       scripts   sys        var
Clients  etc           media  proc            root            smb_mnt   test       vm_disks
data     home          misc   projectfiles    run             software  tmp
dev      known_hosts   mnt    Remote_UserFiles  SampleFiles     srv       tmp_mount
:/#
```

Now reset the root password
#passwd root
Xyz123
Xyz123

Other problem may be
With /etc/fstab
Or
With /etc/default/grub

Now check to run the commands like fdisk (disk related) but I am going to get the error as can't open file : No such file or directory

```
:/# fdisk -l /dev/sda
fdisk: cannot open /dev/sda: No such file or directory
You have new mail in /var/mail/root
:/# _
```

If we see /dev/ contents , you won't see much files that means linux kernel did not started yet and did not probe the hard disks yet.

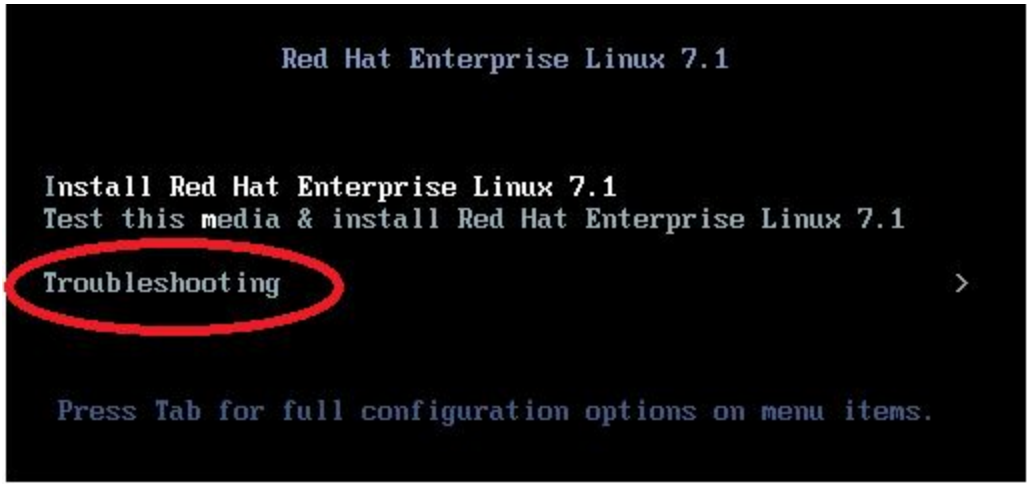
```
:/# ls /dev
null
:/#
```

So if you need to work the problems with **partitioning or File System repair, you can't fix in single user mode or Emergency mode.**

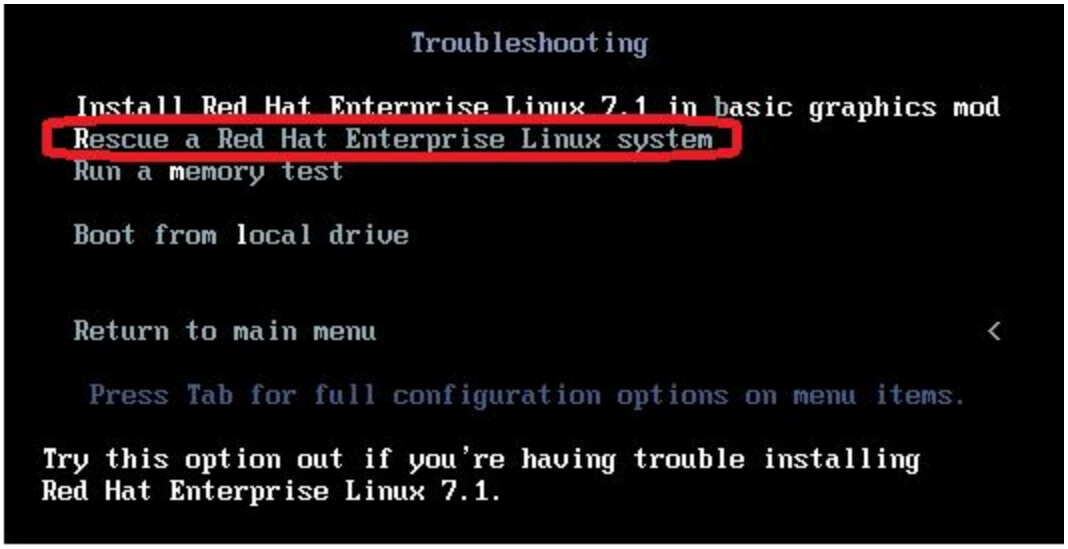
Instead we need work in RESCUE mode

Fixing boot partition issue

By inserting the Linux boot DVD and reboot the server and goes to its menu and select the rescue mode as below



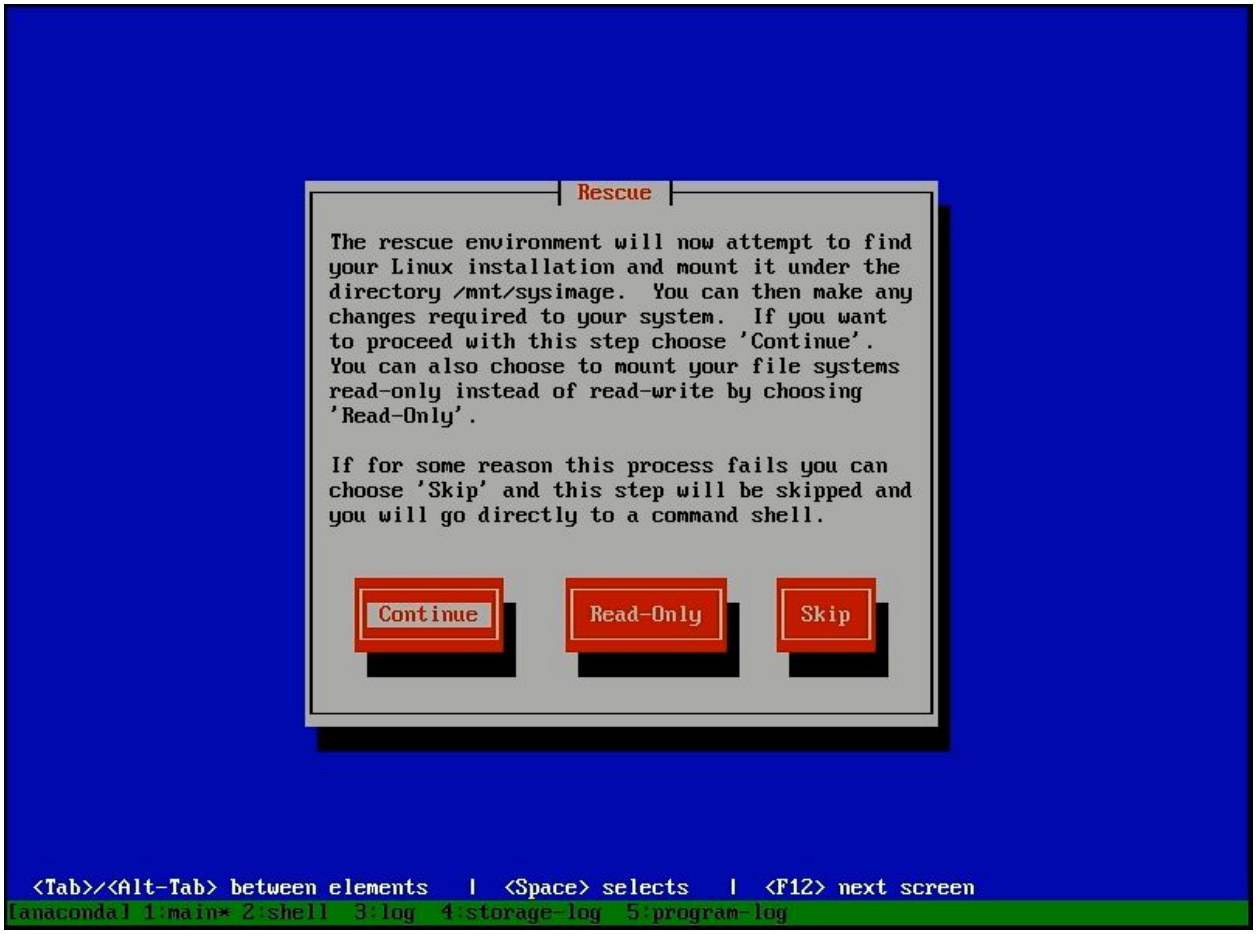
Then go to Rescue mode



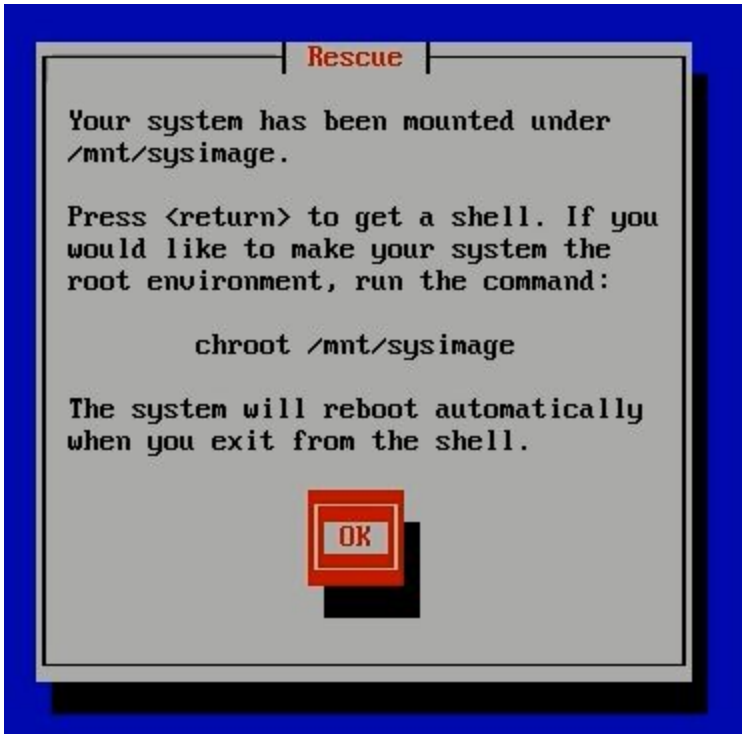
So then it loads the linux kernel and now u can work with things like disk and repair corrupted File System and so on.

Even Though it looks like installation is begin but actually it is not complete installing the product but only loading the rescue environment

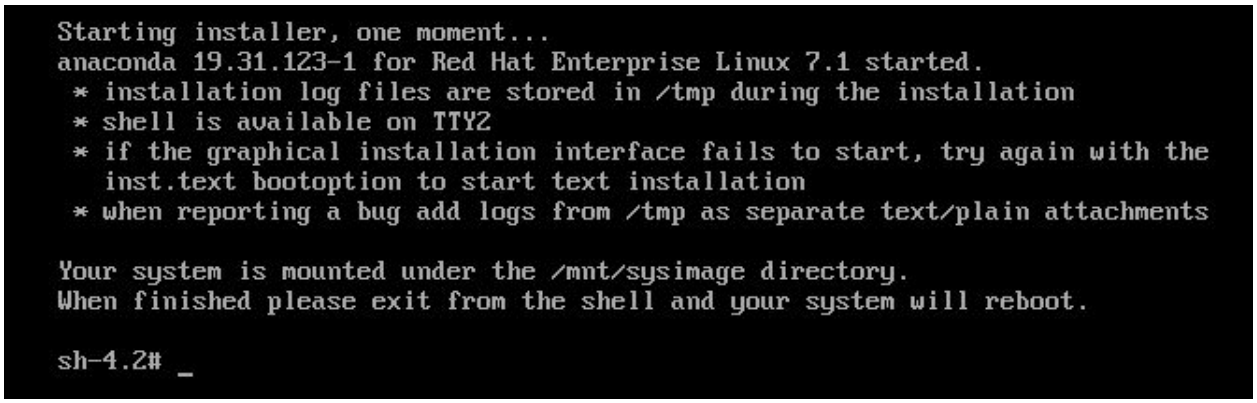
Now I have the below option that tells me about the rescue env (which will attempt to find my linux env)



Hit on Continue, then again one more message comes in



Then click on OK



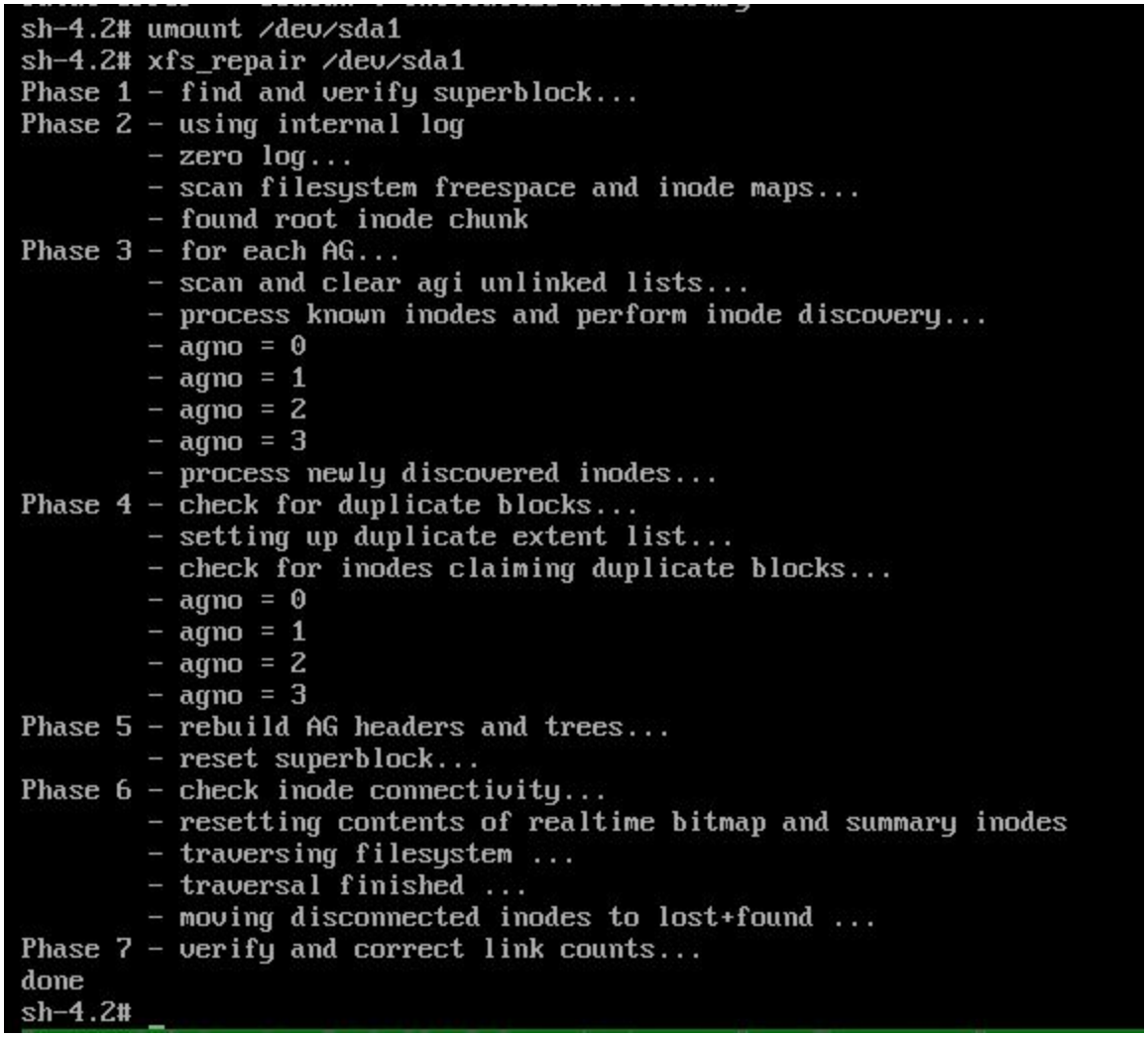
NOW RUN THE COMMAND

#chroot /mnt/sysimage

```
Starting installer, one moment...
anaconda 19.31.123-1 for Red Hat Enterprise Linux 7.1 started.
 * installation log files are stored in /tmp during the installation
 * shell is available on TTY2
 * if the graphical installation interface fails to start, try again with the
   inst.text bootoption to start text installation
 * when reporting a bug add logs from /tmp as separate text/plain attachments

Your system is mounted under the /mnt/sysimage directory.
When finished please exit from the shell and your system will reboot.

sh-4.2# fdisk -l /dev/sda

Disk /dev/sda: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x00096791

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048     1026047      512000   83  Linux
/dev/sda2         1026048    41943039    20458496   8e  Linux LVM
sh-4.2# fsck /dev/sda1
fsck from util-linux 2.23.2
If you wish to check the consistency of an XFS filesystem or
repair a damaged filesystem, see xfs_repair(8).
sh-4.2#
```

Using fsck (in RHEL6) or xfs_repair (in RHEL7),we can repair the File System corrupt issues.

```
sh-4.2# xfs_repair /dev/sda1
xfs_repair: /dev/sda1 contains a mounted filesystem
xfs_repair: /dev/sda1 contains a mounted and writable filesystem

fatal error -- couldn't initialize XFS library
sh-4.2# umount _
```



Recovering from corrupted boot loader

There are some scenarios where grub might got corrupted and you want to recover it or repair it.

This repair can be done by re-installing a new grub from RHEL DVD

To recover the grub , the steps are as below

- Insert the RHEL DVD and make the system boot from CD/DVD
- Boot the system in rescue mode

Check on which partition does your /boot is located using fdisk -l command

Then run the below command to install new grub

#grub-install /dev/sda2

If it shows no error than it means we have successfully recovered the grub

Interview Question

Q1)What is booting process in linux?
Q2)How to go into single user mode?
Q3) suppose if you lost root password , how you will recover?
Q4) how to go into rescue mode?