

VMware



Learn about VMware virtualization for its products like vsphere ESX and ESXi, vCenter Server, VMware View, VMware P2V and many more

Linux



Step by step configuration tutorials for many of the Linux services like DNS, DHCP, FTP, Samba4 etc including many tips and tricks in Red Hat Linux.

Database



Learn installation and configuration of databases like Oracle, My SQL, Postgresql, etc including many other related tutorials in Linux.

ICE ... ITS CALLED TOMORROW !!!

Step by Step Linux Boot Process Explained In Detail

POSTED BY DEEPAK PRASAD MONDAY, MARCH 24, 2014 6 COMMENTS

You can find a number of articles explaining you the Linux Boot Process. Well I am not going to tell any thing different but I would like to show my point of perspective on some extra details of the **Linux Booting procedure**.

The stages involved in Linux Booting Process are:

BIOS

Boot Loader

- MBR
- GRUB

Kernel

Init

Runlevel scripts

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 instruction which tells the CPU to bring 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 GRUB is used with the details of the filesystem in `/boot/grub.conf` and file system drivers

GRUB (GRand Unified Boot loader)

LIKE US ON FACEBOOK FOR ALL UPDATES

POPULAR POSTS

INTERVIEW QUESTIONS ON VMWARE ESXI WITH ANSWERS

STEP BY STEP LINUX BOOT PROCESS EXPLAINED IN DETAIL

10 PRACTICAL EXAMPLES TO USE USERADD COMMAND IN LINUX

HOW TO CHECK THE LOCK STATUS OF ANY USER ACCOUNT IN LINUX

6 COMMANDS TO LIST THE LOGGED IN USERS IN LINUX

HOW TO CREATE USER WITHOUT USERADD COMMAND IN LINUX

HOW TO EXTEND/RESIZE LOGICAL VOLUME AND VOLUME GROUP IN LINUX

HOW TO PRESERVE SYMBOLIC LINKS WITH TAR COMMAND IN UNIX/LINUX

HOW TO CONFIGURE SAMBA 4 AS SECONDARY DOMAIN CONTROLLER

TUTORIAL FOR MONITORING TOOLS SAR AND KSAR WITH EXAMPLES IN LINUX

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

Sample /boot/grub/grub.conf

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.18-194.26.1.el5)
    root (hd0,0)
    kernel /vmlinuz-2.6.18-194.26.1.el5 ro root=/dev/VolGroup00/root
    clocksource=acpi_pm divisor=10
    initrd /initrd-2.6.18-194.26.1.el5.img
title Red Hat Enterprise Linux Server (2.6.18-194.11.4.el5)
    root (hd0,0)
    kernel /vmlinuz-2.6.18-194.11.4.el5 ro root=/dev/VolGroup00/root
    clocksource=acpi_pm divisor=10
    initrd /initrd-2.6.18-194.11.4.el5.img
title Red Hat Enterprise Linux Server (2.6.18-194.11.3.el5)
    root (hd0,0)
    kernel /vmlinuz-2.6.18-194.11.3.el5 ro root=/dev/VolGroup00/root
    clocksource=acpi_pm divisor=10
    initrd /initrd-2.6.18-194.11.3.el5.img
```

For more information on GRUB and LILO follow the below link

[What is GRUB Boot Loader ?](#)

Kernel

This can be considered the heart of operating system responsible for handling all system processes.

Kernel is loaded in the following stages:

1. Kernel as soon as it is loaded configures hardware and memory allocated to the system.
2. Next it **uncompresses** the initrd image (compressed using **zlib** into **zImage** or **bzImage** formats) and mounts it and loads all the necessary drivers.
3. Loading and unloading of kernel modules is done with the help of programs like **insmod**, and **rmmod** present in the initrd image.
4. Looks out for hard disk types be it a LVM or RAID.
5. **Unmounts** initrd image and frees up all the memory occupied by the disk image.
6. Then kernel mounts the **root partition** as specified in grub.conf as **read-only**.
7. Next it runs the **init** process

For more information on kernel follow the below link

[What is a Kernel in Linux?](#)

Init Process

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

Sample output defining the default boot runlevel inside `/etc/inittab`

```
# Default runlevel. The runlevels used by RHS are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:
```

As per above O/P system will boot into **runlevel 5**

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).

Runlevel scripts

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

```
Runlevel Directory
0 /etc/rc.d/rc0.d
1 /etc/rc.d/rc1.d
2 /etc/rc.d/rc2.d
3 /etc/rc.d/rc3.d
4 /etc/rc.d/rc4.d
5 /etc/rc.d/rc5.d
6 /etc/rc.d/rc6.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.
- **For more details on scripts inside `/etc/rc.d` follow the below link**
[What are the s and k scripts in the etc rcx.d directories](#)
- 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.

Follow the below links for more tutorials

[What is the difference/comparison between Unix and Linux ?](#)

[What are the maximum and minimum limits for RHEL 4 vs 5 vs 6 ?](#)

[Step by Step Linux Boot Process Explained In Detail](#)

[What is the difference between ext3 and ext4 filesystem in Linux ?](#)

[How to configure Private Network in VMware Workstation](#)

[10 examples to help you understand top command usage in Unix/Linux](#)

[Configure Red Hat Cluster using VMware, Quorum Disk, GFS2, Openfiler](#)

[Tutorial for Monitoring Tools SAR and KSAR with examples in Linux](#)

[15 tips to enhance security of your Linux machine](#)

[Why is Linux more secure than windows and any other OS](#)

[Understanding Load Average in Linux and when to be worried about it?](#)

[Disk Attachment Technology FC vs SAS vs iSCSI](#)

[Understanding UMASK value in Linux](#)

[How to keep a track of all the commands run by any user in Linux](#)

[How do you check Linux machine is Physical or Virtual remotely?](#)

[RAID levels 0, 1, 2, 3, 4, 5, 6, 0+1, 1+0 features explained in detail](#)

6 comments:



Rajat Sharma 5 Oct 2015, 18:50:00

Precisely explained with good detail thanks

[Reply](#)



Unknown 28 Oct 2015, 22:26:00

Good

[Reply](#)



Unknown 28 Oct 2015, 22:26:00

Good

[Reply](#)



Unknown 9 Jan 2016, 03:52:00

good

[Reply](#)



Unknown 9 Jan 2016, 03:52:00

good

[Reply](#)



kuldeep kamble 13 Jan 2016, 22:33:00

nice.. thanks

[Reply](#)

Enter your comment...

Comment as:

Unknown (Goo ▼)

[Sign out](#)

[Publish](#)

[Preview](#)

☐ [Notify me](#)