**Tutorial No. - 2**

**1.Briefly explain linux booting system?**

**Ans:** The following are the 6 high level stages of a typical Linux boot process.  
1. 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-rom, 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.

2. 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 table info in next 64 bytes 3) mbr validation check in last 2 bytes.
* It contains information about GRUB (or LILO in old systems).
* So, in simple terms MBR loads and executes the GRUB boot loader.

3. GRUB

* GRUB stands for Grand Unified Bootloader.
* 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.
* GRUB has the knowledge of the filesystem (the older Linux loader LILO didn’t understand filesystem).
* Grub configuration file is /boot/grub/grub.conf (/etc/grub.conf is a link to this). The following is sample grub.conf of CentOS.

#boot=/dev/sda

default=0

timeout=5

splashimage=(hd0,0)/boot/grub/splash.xpm.gz

hiddenmenu

title CentOS (2.6.18-194.el5PAE)

root (hd0,0)

kernel /boot/vmlinuz-2.6.18-194.el5PAE ro root=LABEL=/

initrd /boot/initrd-2.6.18-194.el5PAE.img

* As you notice from the above info, it contains kernel and initrd image.
* So, in simple terms GRUB just loads and executes Kernel and initrd images.

4. Kernel

* Mounts the root file system as specified in the “root=” in grub.conf
* Kernel executes the /sbin/init program
* Since init was the 1st program to be executed by Linux Kernel, it has the process id (PID) of 1. Do a ‘ps -ef | grep init’ and check the pid.
* initrd stands for Initial RAM Disk.
* initrd is used by kernel as temporary root file system until kernel is booted and the real root file system is mounted. It also contains necessary drivers compiled inside, which helps it to access the hard drive partitions, and other hardware.

5. Init

* Looks at the /etc/inittab file to decide the Linux run level.
* Following are the available run levels
  + 0 – halt
  + 1 – Single user mode
  + 2 – Multiuser, without NFS
  + 3 – Full multiuser mode
  + 4 – unused
  + 5 – X11
  + 6 – reboot
* Init identifies the default initlevel from /etc/inittab and uses that to load all appropriate program.
* Execute ‘grep initdefault /etc/inittab’ on your system to identify the default run level
* If you want to get into trouble, you can set the default run level to 0 or 6. Since you know what 0 and 6 means, probably you might not do that.
* Typically you would set the default run level to either 3 or 5.

6. Runlevel programs

* When the Linux system is booting up, you might see various services getting started. For example, it might say “starting sendmail …. OK”. Those are the runlevel programs, executed from the run level directory as defined by your run level.
* Depending on your default init level setting, the system will execute the programs from one of the following directories.
  + Run level 0 – /etc/rc.d/rc0.d/
  + Run level 1 – /etc/rc.d/rc1.d/
  + Run level 2 – /etc/rc.d/rc2.d/
  + Run level 3 – /etc/rc.d/rc3.d/
  + Run level 4 – /etc/rc.d/rc4.d/
  + Run level 5 – /etc/rc.d/rc5.d/
  + Run level 6 – /etc/rc.d/rc6.d/
* Please note that there are also symbolic links available for these directory under /etc directly. So, /etc/rc0.d is linked to /etc/rc.d/rc0.d.
* Under the /etc/rc.d/rc\*.d/ directories, you would see programs that start with S and K.
* Programs starts with S are used during startup. S for startup.
* Programs starts with K are used during shutdown. K for kill.
* There are numbers right next to S and K in the program names. Those are the sequence number in which the programs should be started or killed.
* For example, S12syslog is to start the syslog deamon, which has the sequence number of 12. S80sendmail is to start the sendmail daemon, which has the sequence number of 80. So, syslog program will be started before sendmail.

**2.Explain Open source distribution?**

**Ans :**

1.“Open Source” is a software-licensing model where the source code of the software is typically made available royalty-free to the users of the software, under terms allowing redistribution, modification and addition, though often with certain restrictions.

2.The support, training, updates and other services for the software may be provided by a range of entities, increasingly under commercial arrangements. Open source programs are often, though not exclusively, developed through a collaborative effort in which a number of persons contribute elements of the final software. Software companies are also contributing paid programmer time and programs developed in-house to the open source community.

3.Unique about the open source process is that once software has been licensed under an open source license, the collaborative process is no longer tied to a single individual or company. Everyone is free to become a contributor to or distributor of open source software, starting from anyone’s open source software.

4.Open source software is distinguished from most other commercial software because its development frequently takes place collaboratively among many individual developers, working alone or for different companies, without contracts or other formal arrangements among them.

5.In the case of Linux, that open source development project is coordinated by an overall project leader, Linus Torvalds. The Linux team and Torvalds evaluate the quality of contributions they receive from around the world, and they decide whether to include those contributions as a part of Linux.

6. Contributors to open source software can be individuals or companies. Their contributions are combined at the project level with the contributions of other individuals and companies into larger works. Those larger open source works, with their many contributions, are then distributed to the public. Some companies take software distributed by open source projects and aggregate it still further into their own open source products, which they then distribute.

7. A single operating system like Linux, a single web server like Apache, or a single commercial product like a cell phone or a television OpenSource Distribution of Software 45 recorder that includes Linux and Apache may be the result of many contributions by many original authors and distributors along the way.

**3.Explain commercial distribution?**

**Ans :**

1. “Commercial Software” is the model where the software developed by a commercial entity is typically licensed for a fee to a customer (either directly or through channels) in object, binary or executable code. The commercial entity often provides support, training, updates and other similar services needed by customers to efficiently use that software.

2.The source code of the software may be made available1 to certain users of the software through special licensing or other agreements, but is usually Open Source and Commercial Software not distributed to the general public, and may not be copied or modified except in a manner provided for in such agreements.

3. The focus of commercial software providers is on the functionality, features and innovativeness of their technology to meet the customer’s needs, as their revenue model is based on the customer licensing their software.

4.Customers purchase new versions of software when it provides new functionality, features and value. This incentive drives a tremendous flow of research and development spending into new software, the results of which include higher productivity, lower costs of business, and new tools for learning