**Exercise 2 STUDY OF BOOTING PROCESS**

**Date:**

**Question:** Understand the Booting Process of the LINUX /Windows and answer the following questions:

**Linux Boot Process**

Press the power button on your system, and after few moments you see the Linux login prompt.

Have you ever wondered what happens behind the scenes from the time you press the power button until the Linux login prompt appears?

The following are the 6 high level stages of a typical Linux boot process.

**BIOS**

Basic Input/Output System executes MBR

**MBR**

**GRUB**

**KERNEL**

**INIT**

**RUNLEVEL**

Master Boot Record executes GRUB

Grand Unified Bootloader executes Kernel

Kernel executes /sbin/init

Init executes run level programs

Runlevel programs are executed from /etc/rc.d/rc\*.d/

### 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).
* So, in simple terms GRUB just loads and executes Kernel and initrd images.

It 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

### 4. Kernel

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

* 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.

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

### 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.

There you have it. That is what happens during the Linux boot process.

**QUESTIONS**

* + 1. What is meant by booting?
    2. How to check the current runlevel details of your system?

###### Torvalds, Wrote most of the Linux Kernel in C++ programming Language, do you agree?

* + 1. To which runlevel linux boot by default?

### **What is initrd image and what is its function in the linux booting process ?**

### **What are the run levels in linux and how to change them ?**

### **Explain the terms suid, sgid and sticky bit ?**

### **Explain briefly the procedure for re-installing Grub in Linux ?**

* + 1. **How do you boot your system into the following modes, when you are in some trouble ?**
       1. **Rescue mode**
       2. **Single user mode**
       3. **Emergency mode**

**Booting procedure of Windows operating system**

Every day we are using an operating system and most specifically a Windows operating system but most of us are not aware of the process or procedures involved in the starting of Windows. Here we are going to learn the step by step processes involved in the booting procedure of a Windows operating system.

**Functions of BIOS**

The first process starting when you turn on your computer is BIOS i.e, Basic Input Output System. BIOS has two functions, to conduct POST and read MBR.



a) **POST** - POST stands for Power On Self Test. POST checks all the hardware devices connected to a computer like RAM, hard disk etc and make sure that the system can run smoothly with those hardware devices. If the POST is a failure the system halts with a beep sound.

b) Now BIOS checks the **boot priority**. We can set the boot priority as CD drive, hard disk or floppy drive.

c) **MBR** - The next duty of BIOS is to read the MBR. MBR stands for Master Boot Record and its the first sector on a hard disk. MBR

contains the partition table and boot loader.

**Functions of Boot loader**

Now BIOS has passed the control to boot loader and boot loader is a small program which loads kernel to computers memory. Actually there are two stages of boot loaders, stage 1 boot loader and stage 2 boot loader. MBR contains the stage 1 boot loader and stage 1 boot loader is a link to the stage 2 boot loader. The stage 2 boot loader resides in the boot partition and it loads the kernel to memory.

**Boot files and functions**

There are three boot files in a Windows operating system and they are NTLDR, NTDETECT.COM and Boot.ini. The boot files are found in the active partition of hard disk and its normally C drive in a Windows machine.

**NTLDR**- NTLDR stands for NT Loader and its the second stage bootloader. The path of NTLDR is C:\Windows\i386\NTLDR.

**Boot.ini**- Boot.ini contains the configuration files of NTLDR. When the operating system is loaded we cannot pass any arguments to kernal, so those arguments are passed through boot.ini. You can edit boot.ini by opening through notepad. The path of Boot.ini is C:\boot.ini.

**NTDETECT.COM**- This file detect hardware's and passes information to NTLDR. Using the collected information the NTLDR creates a hardware key and this key is used to detect hardware's. A new hardware key is generated after each reboot of the operating system and that's why system asks to reboot after installation of a new hardware. The hardware keys created by NTLDR can be found in Windows registry at HKEY\_LOCAL\_MACHINE -> HARDWARES.

**Kernel and its functions**

After executing the functions of boot files the control is passed to Kernel. ntoskrnal.exe is the kernel file in a Windows machine and its path is C:\Windows\system 32\ntoskrnal.exe. Kernel acts as a layer between software and hardware. The library file hal.dll (C;\Windows\system32\hal.dll) helps Kernel to interact with hardware's. HAL stands for Hardware Abstraction Layer and this hal.dll file is machine specific. Now the drivers for hardware's are loaded from the file C:\Windows\system32\config\system and the Kernel is loaded to primary memory.

**Services and log in procedure**

When kernel is loaded in the primary memory services for each process is started and the registry entry for those services can be found at HKEY\_LOCAL\_MACHINE - System - Current control set - Services.

Winlogon.exe (C:\Windows\system32\winlogon.exe) is the last service started during this process. Winlogon.exe starts the log in procedures of windows machine. It first calls the library file msgina.dll (C:\Windows\system32\msgina.dll).

MSGINA stands for Microsoft Graphics Identification and Authentication and it provides the log in window. Now msginal.dll passes the control to LSA (Local Security Authority), it verifies the username and password from the SAM file. SAM (Security Accounts Manager) contains the information about all users created in a Windows operating system.

Now the booting procedure is over and we have reached the desktop of Windows operating system.

**QUESTIONS**

* + 1. **Which file must be located in the root folder of the Active Partition for Dual Booting of Windows XP and Windows ME to boot?**
    2. **Very frequently, users in your office put floppy disks in the drives and forget to take them out. This results in boot-up problems and results in support requests. How would you stop this computer from looking in the A: drive for boot files first.**
    3. **If, upon Booting up a Computer, you see the error message “Keyboard Error…” What do you think could be the problem?**
    4. **If, upon booting up a PC, you see a message which says “Invalid Drive Configuration..” What do you think is causing this?**
    5. **What is the command to start Windows System Restore from the Command Line?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl No:** | **Assessment Process Description** | | **Mark(s)** |
| 1 | Precise answering | | 5 |
| 2 | Explanation | | 5 |
|  | Total(10) | | 10 |
| Remarks |  | | |
| Date of Completion: | | Signature | |

**Result:**