



Solaris Filesystem

SOLARIS uses **VFS (Virtual File System architecture)**. It provides standard interface for different file system types. The VFS architecture enables kernel to perform basic file operation such as reading, writing and listing. It is called virtual because the user can issue same command to work regardless of the file system. **SOLARIS** also uses **memory based file system** and **disk based file system**.

Lets discuss some memory based file systems:

Memory based File Systems:

It use the physical memory rather than disk and hence also called **Virtual File System** or **pseudo file system**. Following are the Memory based file system supported by **SOLARIS**:

1. **Cache File System(CacheFS)**: It uses the local disk to cache the data from the slow file systems like CD - ROM.
2. **Loopback File System(LOFS)**: If we want to make a file system e.g: /example to look like /ex, we can do that by creating a new virtual file system known as Loopback File System.
3. **Process File System(PROOFS)**: It is used to contains the list of active process in **SOLARIS** by their process ID, in the /proc directory. It is used by the ps command.
4. **Temporary File System(TEMPFS)**: It is the temporary file system used by **SOLARIS** to perform the operation on file systems. It is default file system for /tmp directory in **SOLARIS**.
5. **FIFOFS**: First in first out file system contains named pipe to give processes access to data
6. **MNTFS**: It contains information about all the mounted file system in **SOLARIS**.
7. **SWAPFS**: This file system is used by kernel for swapping.

Disk Based File System:

The disk based file systems resides on disks such as hard disk, cd-rom etc. Following are the disk based file system supported by **SOLARIS**:

1. **High Sierra File System(HSFS)**: It is the file system for CD-ROMs. It is read only file system.
2. **PC File System(PCFS)**: It is used to gain read/write access to the disks formatted for DOS.

3. **Universal Disk Format(UDF)**: It is used to store information on DVDs.

4. **Unix File System(UFS)**: It is default File system used in **SOLARIS**. We will discuss in details below.

Device File System (devfs)

The device file system (**devfs**) manages devices in **Solaris 10** and is mounted to the mount point/devices.

The files in the /dev directory are symbolic links to the files in the /devices directory.

/ root directory
/usr man pages information
/opt 3rd party packages
/etc system configuration files
/dev logical drive info
/devices physical devices info
/home default user home directory
/ kernel Info abt kernel(genunix for Solaris)
lost+found unsaved data info
/proc all active PID's running
/tmp Temporary files system
/lib library file information(debuggers, compilers)
/var It contains logs for troubleshooting
/bin Symbolic link to the /usr/bin directory (Symbolic link is same as shortcut in windows)
/export It commonly holds user's home directory but can customized according the requirement
/mnt Default mount point used to temporarily mount file systems
/sbin Contains system administration commands and utilities. Used during booting when /usr/bin is not mounted.

Important: / is the root directory and as the name suggests, other directories spawn from it.

Linux Filesystem

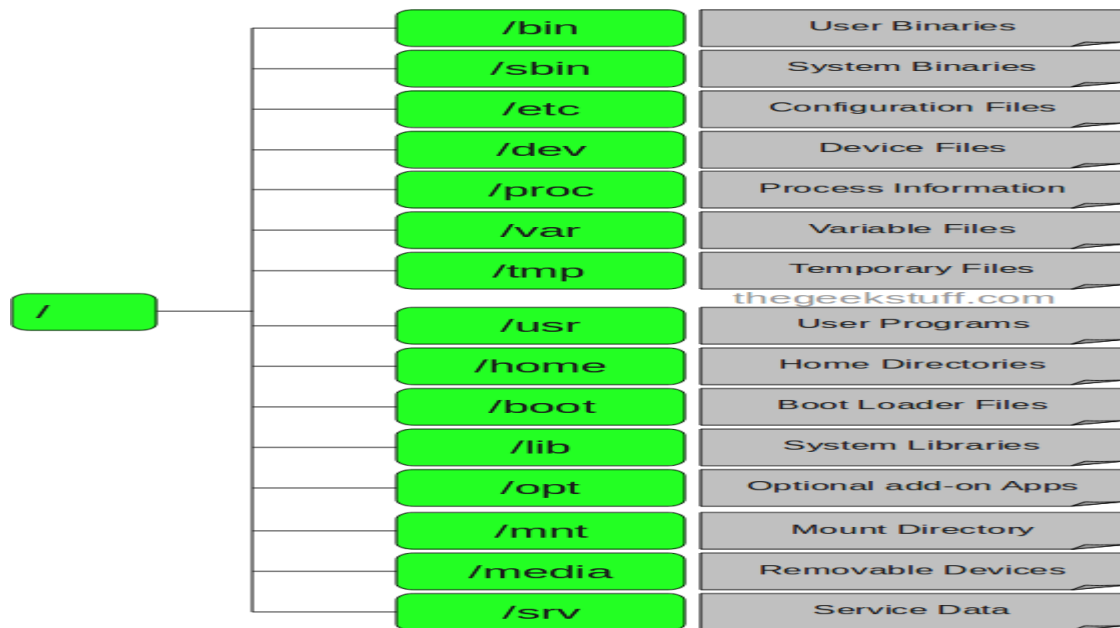
Now below is a very brief comparison of the most common file systems in use with the Linux world.

File System	Max File Size	Max Partition Size	Journaling	Notes
Fat16	2 GB	2 GB	No	Legacy
Fat32	4 GB	8 TB	No	Legacy
NTFS	2 TB	256 TB	Yes	(For Windows Compatibility) NTFS-3g is installed by default in Ubuntu, allowing Read/Write support
ext2	2 TB	32 TB	No	Legacy
ext3	2 TB	32 TB	Yes	Standard linux filesystem for many years. Best choice for super-standard installation.

ext4	16 TB	1 EiB	Yes	Modern iteration of ext3. Best choice for new installations where super-standard isn't necessary.
reiserFS	8 TB	16 TB	Yes	No longer well-maintained.
JFS	4PB	32PB	Yes (metadata)	Created by IBM - Not well maintained.
XFS	8 EB	8 EB	Yes (metadata)	Created by SGI. Best choice for a mix of stability and advanced journaling.
GB = Gigabyte (1024 MB) :: TB = Terabyte (1024 GB) :: PB = Petabyte (1024 TB) :: EB = Exabyte (1024 PB)				

Journaling

A journaling file system is more reliable when it comes to data storage. Journaling file systems do not necessarily prevent corruption, but they do prevent inconsistency and are much faster at file system checks than non-journaled file systems. If a power failure happens while you are saving a file, the save will not complete and you end up with corrupted data and an inconsistent file system. Instead of actually writing directly to the part of the disk where the file is stored, it first writes it to another part of the hard drive and notes the necessary changes to a log, then in the background it goes through each entry to the journal and begins to complete the task, and when the task is complete, it checks it off on the list. Thus the file system is always in a consistent state (the file got saved, the journal reports it as not completely saved, or the journal is inconsistent (but can be rebuilt from the file system)). Some journaling file systems can prevent corruption as well by writing data twice.



1. / – Root

- Every single file and directory starts from the root directory.
- Only root user has write privilege under this directory.
- Please note that /root is root user's home directory, which is not same as /.

2. /bin – User Binaries

- Contains binary executables.
- Common linux commands you need to use in single-user modes are located under this directory.
- Commands used by all the users of the system are located here.
- For example: ps, ls, ping, grep, cp.

3. /sbin – System Binaries

- Just like /bin, /sbin also contains binary executables.
- But, the linux commands located under this directory are used typically by system administrator, for system maintenance purpose.
- For example: iptables, reboot, fdisk, ifconfig, swapon

4. /etc – Configuration Files

- Contains configuration files required by all programs.
- This also contains startup and shutdown shell scripts used to start/stop individual programs.
- For example: /etc/resolv.conf, /etc/logrotate.conf

5. /dev – Device Files

- Contains device files.
- These include terminal devices, usb, or any device attached to the system.
- For example: /dev/tty1, /dev/usbmon0

6. /proc – Process Information

- Contains information about system process.
- This is a pseudo filesystem contains information about running process. For example: /proc/{pid} directory contains information about the process with that particular pid.
- This is a virtual filesystem with text information about system resources. For example: /proc/uptime

7. /var – Variable Files

- var stands for variable files.
- Content of the files that are expected to grow can be found under this directory.
- This includes — system log files (/var/log); packages and database files (/var/lib); emails (/var/mail); print queues (/var/spool); lock files (/var/lock); temp files needed across reboots (/var/tmp);

8. /tmp – Temporary Files

- Directory that contains temporary files created by system and users.
- Files under this directory are deleted when system is rebooted.

9. /usr – User Programs

- Contains binaries, libraries, documentation, and source-code for second level programs.
- /usr/bin contains binary files for user programs. If you can't find a user binary under /bin, look under /usr/bin. For example: at, awk, cc, less, scp
- /usr/sbin contains binary files for system administrators. If you can't find a system binary under /sbin, look under /usr/sbin. For example: atd, cron, sshd, useradd, userdel
- /usr/lib contains libraries for /usr/bin and /usr/sbin
- /usr/local contains users programs that you install from source. For example, when you install apache from source, it goes under /usr/local/apache2

10. /home – Home Directories

- Home directories for all users to store their personal files.
- For example: /home/john, /home/nikita

11. /boot – Boot Loader Files

- Contains boot loader related files.
- Kernel initrd, vmlinuz, grub files are located under /boot

- For example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic

12. /lib – System Libraries

- Contains library files that supports the binaries located under /bin and /sbin
- Library filenames are either ld* or lib*.so.*
- For example: ld-2.11.1.so, libncurses.so.5.7

13. /opt – Optional add-on Applications

- opt stands for optional.
- Contains add-on applications from individual vendors.
- add-on applications should be installed under either /opt/ or /opt/ sub-directory.

14. /mnt – Mount Directory

- Temporary mount directory where sysadmins can mount filesystems.

15. /media – Removable Media Devices

- Temporary mount directory for removable devices.
- For examples, /media/cdrom for CD-ROM; /media/floppy for floppy drives; /media/cdrecorder for CD writer

16. /srv – Service Data

- srv stands for service.
- Contains server specific services related data.
- For example, /srv/cvs contains CVS related data.

Managing Local disk devices (solaris 10)

Disk Naming Convention in Solaris

Hard disk Naming Convention in Solaris

Disk Naming Conventions in SPARC machines (Solaris)

The hard disk in solaris will be referred like

c0t0d0

where c0 stands for controller 0
t0 stands for target 0
d0 stands for disk0

and the first slice will be referred like
c0t0d0s0 where s0 is the first partition

All the disk have entries in /dev/rdisk and /dev/dsk

1./dev/rdisk/c0t0d0s0

2. /dev/dsk/c0t0d0s0

In-order to create file systems or check file systems we normally use /dev/rdisk/c0t0d0s0 .The rdsk directory contains raw devices

In-order to mount the filesystems we normally use /dev/dsk/c0t0d0s0.The dsk directory contains block devices

Hard disk Naming Convention in Solaris

Disk Naming Conventions in x86 machines (Solaris)

The hard disk in solaris will be referred like

c0d0

where c0 stands for controller 0

d0 stands for disk0

and the first slice will be referred like
c0d0s0 where s0 is the first partition

All the disk have entries in /dev/rdisk and /dev/dsk

1./dev/rdisk/c0d0s0

2. /dev/dsk/c0d0s0

In-order to create file systems or check file systems we normally use /dev/rdisk/c0d0s0 .The rdsk directory contains raw devices

In-order to mount the filesystems we normally use /dev/dsk/c0d0s0.The dsk directory contains block devices

To display a physical device name

```
# ls -l /dev/dsk/c0d1s0
```

```
# ls -l /dev/rdsk/c0d1s0
```

Instance names are abbreviated names assigned by the kernel for each device on the system. An instance name is a shortened name for the physical device name.

```
# cat /etc/path_to_inst ==> to view the physical name and instance name of devices
```

```
# prtconf ==> to display the system's configuration information
```

```
# format ==> to display both logical and physical device names
```

Ctrl+d - to exit the format utility

Disk Partition Tables

```
# format
```

```
format> verify ==> to read a disks VTOC (Volume Table of Contents)
```

```
# prtvtoc /dev/dsk/c0d1s0 ==> to read a disks VTOC
```

```
# prtvtoc /dev/dsk/c0d1s2
```

Relabeling a Disk

```
# prtvtoc /dev/dsk/c0d1s2 > /var/tmp/c0d1.vtoc ==> to save a disk's VTOC to a file
```

To relabel a disk

```
# fmthard -s /var/tmp/c0d1.vtoc /dev/rdsk/c0d1s2
```


To initialize the VTOC of a disk

```
# fmthard -s /dev/null /dev/rdisk/c0d1s2
```

```
# fdisk -W - /dev/rdisk/c0d1p0 ==> to view fdisk partition table details
```

```
# fdisk -W /var/tmp/c0d1p0 /dev/rdisk/c0d1p0 ==> to save fdisk partition details to a file
```

```
# fdisk -F /var/tmp/c0d1p0 /dev/rdisk/c0d1p0 ==> to restore fdisk partition table
```

Solaris Management Console

```
# smc & ==> to start Solaris Management Console
```

```
# /etc/init.d/init.wbem status ==> to determine if SMC server is running
```

```
# /etc/init.d/init.wbem stop ==> to stop the SMC server
```

```
# /etc/init.d/init.wbem start ==> to start the SMC server
```

Linux Disk Management

1. View all Disk Partitions in Linux

The following basic command list all existing disk partition on your system. The '-l' argument stand for (listing all partitions) is used with fdisk command to view all available partitions on Linux. The partitions are displayed by their device's names. For example: **/dev/sda**, **/dev/sdb** or **/dev/sdc**.

```
[root@tecmint.com ~]# fdisk -l
```

```
Disk /dev/sda: 637.8 GB, 637802643456 bytes
```

```
255 heads, 63 sectors/track, 77541 cylinders
```

```
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	104391	83	Linux
/dev/sda2		14	2624	20972857+	83	Linux
/dev/sda3		2625	4582	15727635	83	Linux
/dev/sda4		4583	77541	586043167+	5	Extended
/dev/sda5		4583	5887	10482381	83	Linux
/dev/sda6		5888	7192	10482381	83	Linux
/dev/sda7		7193	7845	5245191	83	Linux

```

/dev/sda8          7846          8367      4192933+  82  Linux swap / Solaris
/dev/sda9          8368          77541     555640123+ 8e  Linux LVM

```

2. View Specific Disk Partition in Linux

To view all partitions of specific hard disk use the option **‘-l’** with device name. For example, the following command will display all disk partitions of device **/dev/sda**. If you’ve different device names, simply write device name as **/dev/sdb** or **/dev/sdc**.

```

[root@tecmint.com ~]# fdisk -l /dev/sda

Disk /dev/sda: 637.8 GB, 637802643456 bytes
255 heads, 63 sectors/track, 77541 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1           13       104391   83  Linux
/dev/sda2                14          2624     20972857+  83  Linux
/dev/sda3            2625          4582     15727635   83  Linux
/dev/sda4            4583          77541     586043167+   5  Extended
/dev/sda5            4583          5887     10482381   83  Linux
/dev/sda6            5888          7192     10482381   83  Linux
/dev/sda7            7193          7845      5245191   83  Linux
/dev/sda8            7846          8367      4192933+  82  Linux swap / Solaris
/dev/sda9            8368          77541     555640123+ 8e  Linux LVM

```

3. Check all Available fdisk Commands

If you would like to view all commands which are available for fdisk. Simply use the following command by mentioning the hard disk name such as **/dev/sda** as shown below. The following command will give you output similar to below.

```

[root@tecmint ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
        switch off the mode (command 'c') and change display units to
        sectors (command 'u').

Command (m for help):

```

Type **'m'** to see the list of all available commands of fdisk which can be operated on **/dev/sda** hard disk. After, I enter **'m'** on the screen, you will see the all available options for fdisk that you can be used on the **/dev/sda** device.

```
[root@tecmint ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
        switch off the mode (command 'c') and change display units to
        sectors (command 'u').

Command (m for help): m

Command action
  a   toggle a bootable flag
  b   edit bsd disklabel
  c   toggle the dos compatibility flag
  d   delete a partition
  l   list known partition types
  m   print this menu
  n   add a new partition
  o   create a new empty DOS partition table
  p   print the partition table
  q   quit without saving changes
  s   create a new empty Sun disklabel
  t   change a partition's system id
  u   change display/entry units
  v   verify the partition table
  w   write table to disk and exit
  x   extra functionality (experts only)

Command (m for help):
```

4. Print all Partition Table in Linux

To print all partition table of hard disk, you must be on command mode of specific hard disk say **/dev/sda**.

```
[root@tecmint ~]# fdisk /dev/sda
```

From the command mode, enter **'p'** instead of **'m'** as we did earlier. As I enter **'p'**, it will print the specific **/dev/sda** partition table.

```
Command (m for help): p
```

```
Disk /dev/sda: 637.8 GB, 637802643456 bytes
```

```
255 heads, 63 sectors/track, 77541 cylinders
```

```
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	104391	83	Linux
/dev/sda2		14	2624	20972857+	83	Linux
/dev/sda3		2625	4582	15727635	83	Linux
/dev/sda4		4583	77541	586043167+	5	Extended
/dev/sda5		4583	5887	10482381	83	Linux
/dev/sda6		5888	7192	10482381	83	Linux
/dev/sda7		7193	7845	5245191	83	Linux
/dev/sda8		7846	8367	4192933+	82	Linux swap / Solaris
/dev/sda9		8368	77541	555640123+	8e	Linux LVM

```
Command (m for help):
```

5. How to Delete a Partition in Linux

If you would like to delete a specific partition (i.e **/dev/sda9**) from the specific hard disk such as **/dev/sda**. You must be in fdisk command mode to do this.

```
[root@tecmint ~]# fdisk /dev/sda
```

Next, enter '**d**' to delete any given partition name from the system. As I enter '**d**', it will prompt me to enter partition number that I want to delete from **/dev/sda** hard disk. Suppose I enter number '**4**' here, then it will delete partition number '**4**' (i.e. **/dev/sda4**) disk and shows free space in partition table. Enter '**w**' to write table to disk and exit after making new alterations to partition table. The new changes would only take place after next reboot of system. This can be easily understood from the below output.

```
[root@tecmint ~]# fdisk /dev/sda
```

```
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
        switch off the mode (command 'c') and change display units to
        sectors (command 'u').
```

```
Command (m for help): d
```

```
Partition number (1-4): 4
```

```
Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
You have new mail in /var/spool/mail/root
```

Warning : Be careful, while performing this step, because using option ‘**d**’ will completely delete partition from system and may lost all data in partition.

6. How to Create a New Partition in Linux

If you’ve free space left on one of your device say **/dev/sda** and would like to create a new partition under it. Then you must be in fdisk command mode of **/dev/sda**. Type the following command to enter into command mode of specific hard disk.

```
[root@tecmint ~]# fdisk /dev/sda
```

After entering in command mode, now press “**n**” command to create a new partition under **/dev/sda** with specific size. This can be demonstrated with the help of following given output.

```
[root@tecmint ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
        switch off the mode (command 'c') and change display units to
        sectors (command 'u').

Command (m for help): n

Command action

   e   extended
   p   primary partition (1-4)
e
```

While creating a new partition, it will ask you two options ‘**extended**’ or ‘**primary**’ partition creation. Press ‘**e**’ for extended partition and ‘**p**’ for primary partition. Then it will ask you to enter following two inputs.

- First cylinder number of the partition to be create.
- Last cylinder number of the partition to be created (Last cylinder, +cylinders or +size).

You can enter the size of cylinder by adding “+**5000M**” in last cylinder. Here, ‘+’ means addition and **5000M** means size of new partition (i.e **5000MB**). Please keep in mind that after creating a new partition, you should run ‘**w**’ command to alter and save new changes to partition table and finally reboot your system to verify newly created partition.

```
Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.

The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)

Syncing disks.
```

7. How to Formatting a Partition in Linux

After the new partition is created, don't skip to format the newly created partition using ‘**mkfs**’ command. Type the following command in the terminal to format a partition. Here **/dev/sda4** is my newly created partition.

```
[root@tecmint ~]# mkfs.ext4 /dev/sda4

Create a Mount Point and mount the partition

#mkdir /class    (mount point created)

#mount -t ext4 /dev/sda4 /class (Partition mounted to /class)

#df -h (to display mount points and filesystems)

vi /etc/fstab and add mount filesystem to make it permanent.
```

8. How to Check Size of a Partition in Linux

After formatting new partition, check the size of that partition using flag ‘**s**’ (displays size in blocks) with fdisk command. This way you can check size of any specific device.

```
[root@tecmint ~]# fdisk -s /dev/sda2

5194304
```

9. How to Fix Partition Table Order

If you've deleted a logical partition and again recreated it, you might notice ‘**partition out of order**’ problem or error message like ‘**Partition table entries are not in disk order**’.

For example, when three logical partitions such as (**sda4**, **sda5** and **sda6**) are deleted, and new partition created, you might expect the new partition name would be **sda4**. But, the system would create it as **sda5**. This happens because of, after the partition are deleted, **sda7** partition had been moved as **sda4** and free space shift to the end.

To fix such partition order problems, and assign **sda4** to the newly created partition, issue the '**x**' to enter an extra functionality section and then enter '**f**' expert command to fix the order of partition table as shown below.

```
[root@tecmint ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
        switch off the mode (command 'c') and change display units to
        sectors (command 'u').

Command (m for help): x

Expert command (m for help): f
Done.

Expert command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
```

After, running '**f**' command, don't forget to run '**w**' command to save and exit from fdisk command mode. Once it fixed partition table order, you will no longer get error messages.

10. How to Disable Boot Flag (*) of a Partition

By default, fdisk command shows the boot flag (i.e. '*') symbol on each partition. If you want to enable or disable boot flag on a specific partition, do the following steps.

```
[root@tecmint ~]# fdisk /dev/sda
```

Press '**p**' command to view the current partition table, you see there is a boot flag (asterisk (*)) symbol in orange color on **/dev/sda1** disk as shown below.

```
[root@tecmint ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
        switch off the mode (command 'c') and change display units to
```

sectors (command 'u').

Command (m for help): p

Disk /dev/sda: 637.8 GB, 637802643456 bytes

255 heads, 63 sectors/track, 77541 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	104391	83	Linux
/dev/sda2		14	2624	20972857+	83	Linux
/dev/sda3		2625	4582	15727635	83	Linux
/dev/sda4		4583	77541	586043167+	5	Extended
/dev/sda5		4583	5887	10482381	83	Linux
/dev/sda6		5888	7192	10482381	83	Linux
/dev/sda7		7193	7845	5245191	83	Linux
/dev/sda8		7846	8367	4192933+	82	Linux swap / Solaris
/dev/sda9		8368	77541	555640123+	8e	Linux LVM

Next enter command 'a' to disable boot flag, then enter partition number '1' as (i.e. **/dev/sda1**) in my case. This will disable boot flag on the partition **/dev/sda1**. This will remove the asterisk (*) flag.

Command (m for help): a

Partition number (1-9): 1

Command (m for help): p

Disk /dev/sda: 637.8 GB, 637802643456 bytes

255 heads, 63 sectors/track, 77541 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1		1	13	104391	83	Linux
/dev/sda2		14	2624	20972857+	83	Linux
/dev/sda3		2625	4582	15727635	83	Linux
/dev/sda4		4583	77541	586043167+	5	Extended

/dev/sda5	4583	5887	10482381	83	Linux
/dev/sda6	5888	7192	10482381	83	Linux
/dev/sda7	7193	7845	5245191	83	Linux
/dev/sda8	7846	8367	4192933+	82	Linux swap / Solaris
/dev/sda9	8368	77541	555640123+	8e	Linux LVM

Command (m for help):