

|  |
|--|
| IndianGNU.org - Jul,2007 - GRUB (GRand Unified Bootloader) |
|--|

**Introduction - GRUB (GRand Unified Bootloader)** is the part of GNU project, which was originally designed and implemented by **Erich Stefan Boleyn**. GRUB is one of the best open source **boot** loader used to boot **Linux** operating system. Which can load a wide variety of free operating systems (Linux), as well as proprietary operating systems (like Windows) with **chain-loading**. **LILLO(Linux Loader)** is another boot loader used in Linux. Now we will start with....

### [1] What is Boot Loader? -

**Boot Loader** is the first software program that **runs** when a computer start. It is responsible for **loading** and **transferring** control to an operating system **kernel** software (for e.g. Linux kernel). The kernel, then initializes the rest of the operating system. GRUB understands **filesystems** and **kernel executable formats**, so you can load an arbitrary operating system the way you like, without recording the physical position of your kernel on the disk. Thus you can load the kernel just by specifying its file name and the drive and partition where the kernel resides.

When booting with GRUB, you can use either a **command-line** interface or a **menu interface**. Using the command-line interface, you type the drive specification and file name of the kernel manually. In the menu interface, you just select an OS using the arrow keys. The menu is based on a configuration file which you prepare beforehand. While in the menu, you can switch to the command-line mode, and vice-versa. You can even edit menu entries before using them.

```
root@arunbagul:~# grub ?version
grub (GNU GRUB 0.97)
root@arunbagul:~#
```

**GRUB command-line** (After booting OS)- just type '?grub' on bash command prompt # for grub command-line.

```
root@arunbagul:~# grub
[ Minimal BASH-like line editing is supported.  For
the first word, TAB lists possible command
completions.  Anywhere else TAB lists the possible
completions of a device/filename. ]
```

```
grub>
```

\* Possible GRUB command (just type TAB on grub command-line)

```
grub>
```

Possible commands are: blocklist boot cat chainloader clear cmp color configfile debug device displayapm displaymem dump embed find fstest geometry halt he lp hide impsprobe initrd install ioprobe kernel lock makeactive map md5crypt module modulenounzip pager partnew parttype password pause print quit quiet rea

```
d reboot root rootnoverify savedefault serial setkey setup terminal terminfo testload
testvbe unhide uppermem vbeprobe
```

```
grub>
```

## [2] How to get help about any command line -

```
grub> help root
```

```
root: root [DEVICE [HDBIAS]]
```

Set the current "root device" to the device DEVICE, then attempt to mount it to get the partition size (for passing the partition descriptor in `ES:ESI', used by some chain-loaded bootloaders), the BSD drive-type (for booting BSD kernels using their native boot format), and correctly determine the PC partition where a BSD sub-partition is located. The optional HDBIAS parameter is a number to tell a BSD kernel how many BIOS drive numbers are on controllers before the current one. For example, if there is an IDE disk and a SCSI disk, and your FreeBSD root partition is on the SCSI disk, then use a `1?' for HDBIAS.

```
rootnoverify: rootnoverify [DEVICE [HDBIAS]]
```

Similar to `root', but don't attempt to mount the partition. This is useful for when an OS is outside of the area of the disk that GRUB can read, but setting the correct root device is still desired. Note that the items mentioned in `root' which derived from attempting the mount will NOT work correctly.

```
grub> quit
```

```
root@arunbagul:~#
```

## \*\* Syntax and naming convention used in GRUB -

The device syntax used in GRUB is a bit different from normal syntax. First of all, GRUB requires that the device name be enclosed with "(" and ")".

For example :-

```
(fd0)
```

```
(hd0,1)
```

```
(hd1,3)
```

Here "fd" means floppy disk, "hd" means Hard Disk. **(fd0)**- means first floppy drive , **(hd0,1)** - means first Hard Disk and first partition of first Hard Disk. and **(hd1,3)** - means 2nd Hard Disk and 3rd partition of 3rd Hard Disk. How to use GRUB command-line to boot system, please refer below...

**NOTE** - Drive number and Partition number start from zero(0). Use "tab" to see possible Drive, Partitions no etc.

## [3] How to boot operating systems -

GRUB has two distinct boot methods.

**a) Load an operating system directly,**

**b) Chain-loader ?** load another boot loader which then will load an operating system.

GRUB support **Multiboot**. Linux/Unix like OS can be directly loaded, while other OS which are not supported like Windows can be loaded by using "Chain-loader" method.

**Step(1)- Set root/boot drive -**

\* During GRUB menu you can select grub command-line by pressing "c" and to edit GRUB menu you can press "e"...

```
grub> root (hd0,
```

```
Possible partitions are:
```

```
Partition num: 0, Filesystem type unknown, partition type 0x7
```

```
Partition num: 1, Filesystem type unknown, partition type 0x7
```

```
Partition num: 2, Filesystem type is ext2fs, partition type 0x83
```

```
Partition num: 4, Filesystem type is ext2fs, partition type 0x83
```

```
Partition num: 5, Filesystem type is ext2fs, partition type 0x83
```

```
Partition num: 6, Filesystem type unknown, partition type 0x82
```

```
Partition num: 7, Filesystem type is ext2fs, partition type 0x83
```

```
grub> root (hd0,2)
```

```
grub> cat /grub/menu.lst
```

```
.....
```

```
...
```

\* use this file to find out which will be your possible root partition..

```
grub> cat /
```

```
Possible files are: lost+found System.map-2.6.22-14-generic abi-2.6.22-14-generic
```

```
config-2.6.22-14-generic initrd.img-2.6.22-14-generic.bak memtest86+.bin
```

```
vmlinuz-2.6.22-14-generic initrd.img-2.6.22-14-generic grub System.map-2.6.22-14-xen
```

```
vmlinuz-2.6.22-14-xen config-2.6.22-14-xen xen-3.1.gz initrd.img-2.6.22
```

```
-14-xen initrd.img-2.6.22-14-xen.bak
```

**Step(2) ? Load kernel (don't forget to mention your root partition ie /)-**

```
grub> kernel /vmlinuz-2.6.22-14-generic root=/dev/sda5 ro quiet splash
```

```
grub>
```

**Step(3) How to load ?initrd' -**

```
grub> initrd /initrd.img-2.6.22-14-generic
```

**Step(4) Just Boot your OS -**

```
grub> boot
```

\*\*\* That's it to boot your system from GRUB command line!

**[4] How Boot unsupported operating systems(OS) by using 'chain-loader' method-**

Normally, the boot loader is embedded in the boot sector of the partition on which the operating system is installed. Use '**rootnoverify**' command for setting the correct root device. this command don't attempt to mount that partition.

**Step(1)** Set GRUB's root device to the partition -

```
grub> rootnoverify (hd0,0)
```

**Step(2)** Set the active flag in the partition using 'makeactive' command -

```
grub> makeactive
```

**Step(3)** Load the boot loader with the command chainloader-

```
grub> chainloader +1
```

**Step(4)** Just Boot your OS -

```
grub> boot
```

\* "+1" indicates that GRUB should read one sector from the start of the partition.

**[5] How to read any file from GRUB command-line -**

\* you can easily find out which is the root partition (/) by looking the directory structure!!

```
grub> root (hd0,4)
```

```
grub> cat /
```

```
Possible files are: lost+found var boot home media usr etc cdrom bin dev initrd lib mnt
opt proc root sbin srv sys tmp initrd.img vmlinuz home2 initrd.img.
old vmlinuz.old
```

```
grub> cat /etc/fstab
```

```
# /etc/fstab: static file system information.
```

```
#
```

```
#<file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
```

```

/dev/sda5      /                ext3    defaults,errors=remount-ro    0        1
/dev/sda3      /boot            ext3    defaults                       0        2
/dev/sda8      /home            ext3    defaults                       0        2
/dev/sda6      /usr             ext3    defaults                       0        2

```

```
grub>
```

#### [6] How to Boot DOS/Windows from 2nd Hard Disk -

If you have installed DOS (or Windows) on a non-first hard disk, you have to use the disk swapping technique, because that OS cannot boot from any disks but the first one. Use "map" command. This performs a virtual swap between your first and second hard drive.

```

grub> map (hd0) (hd1)
grub>
grub> map (hd1) (hd0)
grub>

```

NOTE: This is effective only if DOS (or Windows) uses BIOS to access the swapped disks. If that OS uses a special driver for the disks, this probably won't work.

\*\* Another problem arises if you installed more than one set of DOS/Windows onto one disk, because they could be confused if there are more than one primary partitions for DOS/Windows. Certainly you should avoid doing this, but there is a solution if you do want to do so. Use the partition hiding/unhiding technique. If GRUB **hides** a DOS/Windows partition, DOS/Windows will ignore the partition. If GRUB **unhides** a DOS/Windows partition, DOS/Windows will detect the partition.

```

grub> help hide
hide: hide PARTITION
Hide PARTITION by setting the "hidden" bit in its partition type code.

```

```

grub> root (hd0,
Possible partitions are:
Partition num: 0, Filesystem type unknown, partition type 0x7
Partition num: 1, Filesystem type unknown, partition type 0x7
Partition num: 2, Filesystem type is ext2fs, partition type 0x83
Partition num: 4, Filesystem type is ext2fs, partition type 0x83
Partition num: 5, Filesystem type is ext2fs, partition type 0x83
Partition num: 6, Filesystem type unknown, partition type 0x82
Partition num: 7, Filesystem type is ext2fs, partition type 0x83

```

```
grub> hide (hd0,2)
```

```

grub> root (hd0,
Possible partitions are:
Partition num: 0, Filesystem type unknown, partition type 0x7

```

Partition num: 1, Filesystem type unknown, partition type 0x7  
Partition num: 2, Filesystem type **unknown**, partition type 0x93  
Partition num: 4, Filesystem type is ext2fs, partition type 0x83  
Partition num: 5, Filesystem type is ext2fs, partition type 0x83  
Partition num: 6, Filesystem type unknown, partition type 0x82  
Partition num: 7, Filesystem type is ext2fs, partition type 0x83

grub>

grub> **help unhide**

unhide: unhide PARTITION

Unhide PARTITION by clearing the "hidden" bit in its partition type code.

grub>

For example ... see below

grub> **unhide (hd0,0)**

grub> **hide (hd0,3)**

grub> **rootnoverify (hd0,0)**

grub> **chainloader +1**

grub> **makeactive**

grub> **boot**

Thank you,

Arun Bagul

### **Similar Posts:**

- [What's new in GRUB-2?](#)
- [Ubuntu ? GRUB splash image as background for GRUB menu](#)
- [How to create edit/extract initrd in Ubuntu/Debian and Redhat/Fedora Linux ?](#)
- [How to read MBR \(Master Boot Record\)](#)
- [How to disable core\(s\) of CPU](#)
- [Xen virtualization on CentOS linux](#)
- [Single user mode ? How to reset\(bypass\) root password without bootable CD](#)

**Share and Enjoy:**

•

•

•

•

•

•

•

•

•

•

•

•

•