



KYLE RANKIN

# Break In Your Boots

New boots, new bootloaders—both take some time to get used to. In this column, find out all the uncomfortable changes in the new GRUB2 bootloader.

**A few months ago**, I had to replace my favorite pair of shoes: black, suede Converse One Stars (the classic style with no white rubber toe cap, thank you). I had worn the shoes for years, but although most of the shoe held up fine, I had completely worn down the heel. Now, I'm not one to throw away a comfortable pair of shoes. This pair was on its ninth or tenth Shoe-Goo repair, but it finally became hopeless. They had to be replaced. It seemed like a simple task—after all, these shoes had been available all through my adult life, but wouldn't you know, the moment I need another pair, Converse discontinued the model and replaced it with a canvas version with the Chuck Taylor-esque white rubber toe. I had to find a new shoe.

Let me tell you, once you have found the perfect sneaker, it's impossible to find a replacement. Everything I looked at was held up to the standard of the shoe I couldn't have. After a month or two, I finally found shoes that were up to the task, and although I like them, I still miss my old shoes. (oh, wouldn't you know it, now that I bought a replacement, Converse has re-released the One Stars how I like them).

I really should be used to this feeling. It seems every few years some open-source project decides to throw away an entire codebase and start from scratch. Although GNOME and KDE have stirred the pot the most with this, I've also lived through the same thing with the Enlightenment Project, the SysV init to Upstart transition, the LILO bootloader being phased out for GRUB, and now GRUB being replaced by GRUB2. For those of you who thought the difference between GRUB2 and GRUB1 was "one", you are: good at subtraction, a bit of a smart aleck and in for a rude awakening. In this article, I'm going to help you break in your new GRUB2 bootloader, so hopefully some day, it will be as comfortable to you as the original GRUB.

## Why?

The first question you might ask is why we need a new bootloader at all? What is wrong with GRUB? The answer, according to the GRUB2 developers, is that the original GRUB codebase was rather old and had become unmaintainable. The software continued to get new feature requests (such as supporting new hardware and platforms) that ultimately were beyond the scope of the original code, so the decision was made to scrap it and start from scratch. Because it was a complete rewrite, the developers decided to take the opportunity to make a clean break and

redesign the layout and syntax of the configuration files. Along with these changes, GRUB2 has been able to add new features, such as a rescue mode, enhanced graphical menu and splash screen support, full support for UUIDs, and support for non-x86 platforms, such as PowerPC.

## Old Boots

Before I go into how GRUB2 has changed things, I'm going to give a quick overview of GRUB1 (or GRUB Legacy, as they are calling it now) to help highlight the changes for those of you who might be unfamiliar with either bootloader. GRUB (and LILO before it) has been the standard bootloader used by the majority of Linux distributions. When you boot your computer and see a menu that lets you choose between different Linux kernels, or between different versions of Linux and Windows in a dual-boot scenario, you probably are using GRUB. GRUB's job is to allow you to choose between one or more operating systems at boot time and then either load the respective kernel and initrd into memory and start the rest of the boot process, or launch the boot code for some other operating system, such as Windows.

GRUB is quite configurable and organizes itself into a few core programs and directories:

- `/boot/grub/menu.lst`: this is the default configuration file for GRUB, although on some distributions it is a symlink to `/boot/grub/grub.conf`. All of GRUB's configuration is in this file, and users edit this file directly to change any GRUB options.
- `/usr/sbin/grub`: this is the core GRUB binary that you can use (if you learned all of the syntax) to install GRUB onto your system. The syntax is a bit tricky though, so ultimately, other programs appeared to help automate the process.
- `/usr/sbin/grub-install`: this program acts as the front end to `/usr/sbin/grub` and makes it much simpler to install GRUB to your hard drive.
- `/usr/sbin/update-grub`: this script helps automate configuration of the `menu.lst` file. Instead of having to add new kernels to `menu.lst` manually, you can run this script, and it will detect kernels available on your system and build the `menu.lst` for you. In addition, this script can read special configuration options in the comments of `menu.lst` and further

automate the process of providing rescue modes, memtest86+ and other customizations of the file.

Another great feature of GRUB is the fact that even with all of this configuration, if you make a mistake, GRUB allows you to change essentially any configuration option from the boot prompt. At the GRUB menu, you can press the Esc key to change boot options on the fly.

### New Boots

Now that you have learned (or refreshed your memory) about GRUB, you may promptly forget it, because much of what I said above has changed. For starters (and this will drive you crazy), the default key to edit GRUB2 options at boot is Shift, not Esc. Second, the main configuration file has been changed from /boot/grub/menu.lst to /boot/grub/grub.cfg. Not only has the filename changed, but also the syntax inside the file is quite different from what you'd find in menu.lst.

While I'm on the subject of syntax changes, a crucial syntax change that GRUB2 makes is in how it numbers partitions. Where in GRUB your partitions were counted starting from zero, now the count starts with one. To make it more confusing, disk devices still are being counted starting from zero. Confused yet? In short:

```
■ GRUB1: /dev/sda1 = (hd0,0)
```

```
■ GRUB2: /dev/sda1 = (hd0,1)
```

Here's a sample stanza from GRUB's menu.lst and a similar stanza from GRUB2's grub.cfg, so you can compare their syntax:

```
GRUB:
```

```
title      Ubuntu karmic (development branch), kernel 2.6.31-14-generic
uuid      c7b6836f-ac57-47ed-9e7c-b16adb88abed
kernel    /boot/vmlinuz-2.6.31-14-generic root=UUID=
          ↪c7b6836f-ac57-47ed-9e7c-b16adb88abed ro quiet splash
initrd    /boot/initrd.img-2.6.31-14-generic
quiet
```

GRUB2:

```
menuentry 'Ubuntu, with Linux 2.6.31-20-generic' --class ubuntu
  ↪--class gnu-linux --class gnu --class os {
    recordfail
    insmod ext2
    set root='(hd0,2)'
    search --no-floppy --fs-uuid --set
```



visit us at [www.siliconmechanics.com](http://www.siliconmechanics.com)  
or call us toll free at 866-352-1173



As the Chief Financial Officer for Silicon Mechanics, Steve is an Expert where value is concerned. That's why he's pictured here with the Rackform iServ R143.

The R143 is a flexible and affordable 1U server. It features an Intel® Xeon® Processor 3400 Series, with powerful features like Turbo Boost, Hyper-Threading, and DDR3 memory. This processor is also available in a low-voltage version, which can optimize power usage and help contain energy costs. With 6 DDR3 DIMM sockets, 2 Gigabit Ethernet adapters, a PCIe expansion slot, and 4 hot-swap SAS/SATA drive bays, the R143 can handle a lot more than entry-level workloads. With a price that starts around \$1250, you don't have to be a CFO to understand the value.

When you partner with Silicon Mechanics, you get more than a flexible, affordable entry-level server — you get an Expert like Steve.

For more information about the  
Rackform nServ R143  
visit [www.siliconmechanics.com/R143](http://www.siliconmechanics.com/R143)

# Expert included.

```

└─c7b6836f-ac57-47ed-9e7c-b16adb8abed
linux /boot/vmlinuz-2.6.31-20-generic
└─root=UUID=c7b6836f-ac57-47ed-9e7c-b16adb8abed
└─ro quiet splash
initrd /boot/initrd.img-2.6.31-20-generic
}

```

Before you sit down and study the new syntax, I should point out that you are actively discouraged from editing `grub.cfg` directly. This file is generated from a series of scripts and configuration files I will talk about later, so any changes you make will be overwritten the next time any package updates that would trigger the GRUB2 update.

Like with GRUB, here are the core files and programs involved in GRUB2 configuration:

- `/boot/grub/grub.cfg`: this is the core GRUB2 configuration file but is not to be edited directly.
- `/etc/default/grub`: this is the main configuration file for end users to edit. In this file, you can configure a limited subset of GRUB2 options, such as timeouts, basic kernel boot options and whether to use a graphical console or UUIDs. Every time you make a change to this file, you must run `/usr/sbin/update-grub` for the changes to be reflected. Here are some sample lines from the file to give you some idea of the syntax:

```

GRUB_DEFAULT=0
GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="lsb_release -i -s 2> /dev/null || echo Debian"
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

```

- `/etc/grub.d/`: this directory contains a series of scripts that are executed in numerical order by the `grub-mkconfig` program and will configure different parts of `grub.cfg`. On a default Ubuntu Lucid install, for instance, you would find the following files:

```

$ ls /etc/grub.d/
00_header      10_linux      30_os-prober  README
05_debian_theme 20_memtest86+ 40_custom

```

The order in which configuration options appear in `grub.cfg` is governed by the order its script appears in this directory. So if, for instance, you wanted to have a different OS appear before the Linux options in the menu, you could name the script `01_others`. Although the scripts that are currently there do a few complicated things, essentially your script needs to output the configuration information you want in the proper `grub.cfg` syntax, so I suppose it even

could be a series of echo statements in a shell script. If you want to create a custom configuration script though, Ubuntu has provided the `40_custom` script for you to use that will not risk being overwritten.

- `/usr/sbin/grub-install`: like with GRUB, the GRUB2 `grub-install` program is the recommended way to install GRUB2 onto a device. It calls a number of other scripts that perform various system checks, device probes and everything else that's necessary to install GRUB2 to a boot device.
- `/usr/sbin/update-grub`: this script still exists and is still the recommended way to update GRUB2's configuration file, but now this is a very short shell script that executes `grub-mkconfig`. Whenever you edit a configuration file or script, run this command with no arguments to rebuild the `grub.cfg` file.
- `/usr/sbin/grub-mkconfig`: this program does the real heavy-lifting to build your `grub.cfg` file. It is the program that executes the various scripts in `/etc/grub.d`.

In addition to the above changes, here are a few extra things that are different in GRUB2:

- GRUB2 no longer has stage 1.5 in the boot process.
- On a new Ubuntu install when no other OSes are present, GRUB2 will not display a menu at boot time and will instead boot directly into the Ubuntu install.
- To reiterate, hold Shift instead of Esc to change GRUB2 boot options.

If you are like I was when I first discovered all of these changes, right about now you are feeling like the ground has been moved out from under you. I felt much like I did when I couldn't buy a new pair of black suede One Stars. All I can hope for is that over time, like my new shoes, the uncomfortable parts of GRUB2 will break in, and I will feel comfortable with them and maybe some day even like them as much as GRUB1. ■

Kyle Rankin is a Systems Architect in the San Francisco Bay Area and the author of a number of books, including *The Official Ubuntu Server Book*, *Knoppix Hacks* and *Ubuntu Hacks*. He is currently the president of the North Bay Linux Users' Group.

## Resources

GRUB Wiki: [grub.enbug.org](http://grub.enbug.org)

Ubuntu GRUB Community Documentation:  
<https://help.ubuntu.com/community/Grub2>