

Grub Patch

Carlo Contavalli <ccontavalli at commedia.it>

Thu May 1 17:35:15 CEST 2003

This file documents a patch I've made for grub, the "Grand Unified Boot loader" to be able to specify parameters to the "savedefault" command and to add the "showdefault" command, both in the boot command line and in the grub shell. This manual, the mentioned patch and all the provided files are copyright © Carlo Contavalli 2001-2003. Please read the following sections for more details. **Note that this is free software and authors hold no responsibility for any damage or loss, direct or indirect, caused by using this software. Use it only on your OWN risk and AFTER carefully reading this documentation.** The latest version of this document can be found at <http://www.commedia.it/ccontavalli/>.

1 License, copyright and...

This document and the grub patch described here were written by Carlo Contavalli <ccontavalli at commedia.it> and are thus Copyright © Carlo Contavalli 2002-2003.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts and no Back-Cover Texts.

Any example of program code available in this document should be considered protected by the terms of the GNU General Public License.

This patch is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This patch is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Trademarks are owned by their respective owners.

2 About this document & this patch...

Ok, so why bothering for being able to specify *some more parameters* to "savedefault" and asking for a "showdefault"?

Well, as you probably already know, grub provides the ability to remember the last entry we used to boot and to automatically use it on the next reboot.

A while ago, I was looking for a nice, clean and reliable way to upgrade the kernel of a remotely installed machine I don't have physical access to.

Let's see... we need to compile the new kernel, possibly test it, transfer it on the remote server and install it.

This is not quite an error free procedure: human errors may happen, the compiler or some system library may just be sick when compiling the kernel and our test might just be not enough to ensure the kernel **will work at the first shot on the remote server**.

Even if I use something like "make-kpkg" I may just forgot to apply some fundamental patch or to disable/enable some brand new kernel feature.

The purpose of this patch is to provide a nice way to have more than “one shoot” when trying to upgrade a remotely installed linux kernel.

So, if something bad happens, the system can be rebooted and the error corrected without having to “be there”.

3 The big picture

Grub already has the ability to remember *which kernel to run the next time the system is booted*. You simply need to use the “savedefault” command in the entries you are interested to in “menu.lst” and to add a “default saved” at the beginning of the same file.

However, let’s suppose we just installed a brand new kernel. We could just add a new entry in “menu.lst” and set the default to the number of the new kernel.

The next time we reboot we could just hope that everything will be fine and wait for ssh to come back to life.

But, what would it happen if we were able to put a “savedefault number_of_old_kernel” in the new kernel entry? During the first reboot, the old kernel would be saved as the default one. Can you see the advantage? I still have troubles... now, let’s say you use “default saved” instead of “default new_kernel”, and let’s say you run the grub shell from your linux prompt and give a “savedefault new_kernel”. What about it now? The first time you reboot, the new kernel will be booted (it is the default entry). While the new kernel is booted, “savedefault number_of_old_kernel” will ask grub to run the old kernel next time the system is booted. Now if something goes wrong, we could just reboot the system to have the server back to life with the old kernel. This could be enough for most people: in most data centers, you just need to phone some nice guy to have your server rebooted.

However, the linux kernel gives us a nice hand: if we specify something like “panic=10” to the kernel boot line, linux will reboot in 10 seconds if a panic shows up, and a panic is something you will probably get if the compiler was sick while compiling the kernel or if you forgot some patch or some important driver.

During the first successful boot, we could even add a script in init.d to tell grub to remember to boot the new kernel instead of the old one, and to remove the lines that provided us some more reliability.

4 Changes to grub

In short, this patch adds:

- “showdefault” command, which can be used both from the boot time shell and from the command line interface on your running system. This command will print on the screen the default being currently saved in the boot area.
- “savedefault” is modified to:
 - accept one more parameter at the boot time shell. If no parameter is specified, savedefault will behave in the standard way. If one parameter is specified and this parameter is a number, like in “savedefault 3”, grub is told to remember to boot the entry number 3 instead of the current one.
 - accept two more parameters at the command line interface when run from your command prompt. The first parameter must be the entry we want grub to boot on the next reboot. The second parameter is the name of the stage2 image used to boot. If no image is specified, “savedefault” will try to figure it out by itself, so, no need to worry about it on most installations.

4.1 Installing the patch

This patch was made against the version of grub provided in the Debian archive, namely 0.93 + cvs 20030224. It should also work with grub-0.92 and with grub patched for the el-torito hd boot image, available on the internet in a file named “grub-0.92.a_cd+ef.diff.gz” (this is the patch needed by mkbimage). At most you will get a reject in stage2/builtins.c, since patch is not able to understand where to add the line “&builtin_showdefault,” in builtin_table[]. Since you are a human being, you should be able to insert it in the correct place. Just look for “builtin_table” in your builtins.c and add the line I told you about on any line by itself as long as the alphabetical order of the listed commands is maintained.

This patch is just one morning effort, so, don’t blame me too much if you have troubles. It actually took me longer to write this document than to code the patch. If you have troubles using it, I’d be glad to give you as much help as you may need. Just mail me at <ccontavalli at commedia.it>.

To use the patch, you just need a standard:

```
$ cd grub_sources
$ cat path_to_my_patch/grub-patch-x.y.diff |patch -p1
```

4.2 Using the patch

After you install your brand new kernel, open your “menu.lst” and:

1. Add an entry for your new kernel
2. At the beginning of your configuration file, change any “default number” you may have in “default saved”.
3. In your new kernel entry, add the line “savedefault number_of_old_and_good_kernel_entry”.
4. From the command line, run the grub shell with the command “grub” (you need to be root) and type the command “savedefault number_of_new_kernel_entry”.
5. Check the operation was successful by using “showdefault”.
6. Add the parameter “panic=10” to your kernel image.
7. You should now be done.

This process is not “idiot proof” and does not guarantee against many kernel malfunctions. Anyway, this process can be easily automated and quite a number of kernel problems will be nicely caught and handled without having to access the server physically. At worst, you will be able to make a phone call and ask for somebody to just reboot your server.