

Making MINIX multi-boot compliant

GSoC project information, April 1st 2010

Abstract

Currently, MINIX 3 has its own boot loader, but it is quite old and has some serious limitations. We would like to change the booting structure to use a modern boot loader such as GRUB 2 (without chainloading). To do this project, you should have experience working with booting, for example, writing the boot code for an embedded system. You should also be a very good at writing C and assembly code.

Project description

MINIX comes with its own bootloader to load its kernel and essential servers from disk into memory and execute them. Currently, this is the only way to boot MINIX. If one wants to use MINIX together with other operating systems in a multi-boot configuration, the bootloader offering the choice between operating systems has to invoke the MINIX bootloader to load MINIX (a process called chainloading).

We would like you to make MINIX bootable from one more bootloader, namely GRUB 2. This bootloader defines the multi-boot specification, which allows it to boot all operating systems implementing it directly (that is, without chainloading). If MINIX is made multi-boot compliant, it will be possible to add multiple MINIX kernels and configurations to the GRUB menu. This avoids having to go through multiple boot menus and having to configure them separately. It would also make it easier to port MINIX in the future, as bootloaders are highly platform-specific and GRUB has already been ported to many platforms. Moreover, it would lift restrictions imposed by the MINIX bootloader, such as the inability to load to memory above 16MB. Finally, hypervisors bugs sometimes break the MINIX bootloader. This would be less likely for a widely used bootloader such as GRUB.

This project involves modifying the MINIX boot code to be multi-boot compliant (while retaining support for the old bootloader) as well as writing a module for GRUB 2 to allow it to read the MINIX filesystem. The MINIX image must be organized in a way that GRUB can understand while MINIX must be capable of running in the environment that GRUB creates for it and of parsing the information (boot parameters) GRUB provides to it. For this project you need experience with low-level systems programming on the i386 architecture and in particular assembly language skills.

Requirements

The following is a tentative list of requirements for this project. Except where otherwise noted, these requirements are not (yet) cast in stone and there can be good reasons to change them. These points are more or less ordered by decreasing priority.

- It is an absolute requirement that the MINIX boot monitor keeps working. If changes need to be made, they should be minimal. There is no need (or desire) to make the MINIX boot monitor multiboot compliant (just like the example that was mentioned, Lilo is not multiboot compliant but can still be used to load Linux).
- The multi-boot implementation must support passing boot configuration parameters to MINIX as it does with the boot monitor;
- The primary focus is on GRUB 2, which must be capable of booting a default MINIX installation (for now that probably means sub-partitioned MFS) without chainloading;
- The multi-boot implementation must overcome the limitations of the MINIX boot monitor, such as the inability to load large images;
- The multi-boot standard must be properly implemented and other multi-boot compliant boot monitors should also be able to boot MINIX, at least on Ext2;
- Multi-boot and boot monitor support should be available using the same boot image, which means that for now the a.out format should be supported. Supporting the build system branch of MINIX is not enough. It should be possible to build it on trunk using the default compiler, assembler and linker (ACK).
- It would be nice if the grub configuration tools on Linux can automatically detect MINIX and set up suitable parameters;
- It would be nice to be able to CD-ROM boot or network boot MINIX using GRUB;
- It would be nice if MINIX setup would be able to add itself to a GRUB configuration file on a different Ext2 partition;
- Being able to install GRUB on a MINIX partition does not have a high priority; it is assumed that a MINIX-only system will probably use the boot monitor.

Besides these functional requirements, design and code style should be sufficiently clean to be acceptable for inclusion in MINIX and GRUB following the style guidelines for both projects. This also means that it is preferable that changes are kept small and concentrated to the extent possible.

Original location of this text:

http://groups.google.com/group/minix3/browse_thread/thread/7c827cb363c89346#f92f04169356fa86

Sheep/goat project

As announced on the GSoC 2010 project ideas page (<http://www.minix3.org/soc-2010/>) we intend to select the best possible students for this year's summer of code. There are generally many applicants for very few slots, so we expect that applicants make some effort to show that they are indeed very good.

The project described below is one example of a "sheep/goat" project that allows you to show your ability. It is aimed in particular at the more low-level projects, such as the multi-boot project.

Currently, MINIX runs well on much hardware as well as on many emulators. However, it fails to boot on VirtualBox 3.1 (it worked on 3.0, see <http://wiki.minix3.org/en/UsersGuide/RunningMinixOnVirtualBox>). We have solved similar issues on other emulators (for example see

http://groups.google.com/group/minix3/browse_thread/thread/27ee96b478... about KVM). Generally some part of the emulation is broken (that is, different from the behaviour of the real hardware). Finding out what the problem is is generally not very difficult, but requires good low-level debugging skills as well as an understanding of the boot process.

The challenge is to find out what the problem is that prevents MINIX from running on VirtualBox 3.1. You would impress us if you could provide a fix or workaround for MINIX and/or recommendations to VirtualBox for fixing the problem. However, steps in the right direction such as finding out more details about the situation in which the problem occurs, describing how you debugged the problem and showing what tests you did to verify your hypotheses regarding the problem also give us some indication of your ability.

If you have solutions, ideas or questions regarding this project, please post them as replies in this thread. Feel free also to comment on other's posts, it may be possible to come to a solution through collective work.

This is just an example, please feel free to think of ways to show your ability. If this project is too difficult for you, don't despair; it is aimed specifically at the lower level projects and you may wish to apply for a higher-level project.

Original location of this text:

http://groups.google.com/group/minix3/browse_thread/thread/301d6b262829ed03