

Zack's Kernel News

■ Bug Hunter

The effort to track Kernel bugs via a semi-automated system continues to pose thorny problems for developers. Martin J. Bligh and others continue to maintain the Bugzilla server, and many developers continue to claim bugs and work on them. Over the course of December 2002, it became clear that not all developers were happy with the new system.

John Bradford in particular, decided to write an entirely new bug tracking system on his own. His idea is to create a system that is specific to the Linux kernel, rather than being a generic bug database.

By the end of 2002 he had already completed an initial version of his system, and had put it up at <http://grabjohn.com/kernelbugdatabase/> (Note that to log in as a guest, you need to use the username "guest" and the password "guest"). The basic theory is to automate

the system as much as possible, so that developers can focus on their particular areas of the kernel very quickly, and not waste a lot of time navigating the system by hand. The early stages of the program still have restrictions, such as having to email John directly for a personal account.

There is also some controversy over the whole idea of writing a new program, rather than simply modifying existing systems such as Bugzilla itself. John's answer to this has been that it's easier to start from scratch than to wade through so much existing code that would have to be changed.

From the initial stirrings of his project (which still appears to have no official name), it seems clear that Bugzilla has a good head-start, so John's replacement will probably have to show a big improvement if it is ever going to take its place completely. ■

■ User Limits

Linux continues to be good fodder for school projects. Over the years, many programmers have chosen to implement new algorithms or rewrite whole subsystems, to satisfy their college requirements..

Martin Waitz is one of these. Due in January 2003, his project involved creating a resource container that would allow sysadmins to control access to various resources, not just on a per-user basis, but according to any set of policies they could devise.

The subject came up on the linux-kernel mailing list when Frederik Dannemare asked if there was any way to limit the amount of CPU a given user could use at a given time. Martin offered his project as one method of doing this, but apparently he is not the only one interested in this sort of thing, and various patches to try to solve this problem have been floating around for a long time.

A patch from Connectiva against 2.4 had been forward ported to 2.5, and Karol Golab had a small patch to provide a similar service. Within a day of his initial question, Frederik was drowning in patches to limit CPU usage. Even Martin offered to send his unfinished school project to anyone who was interested.

The hunt for a good per-user (or per-anything) resource control system is very important. Without it, there are fairly trivial ways for any user to bring a Linux system to its knees.

As far back as July 2000, Marcelo Tosatti predicted that decent per-user resource limits would make it into the 2.6 kernel, and then be back-ported into the 2.4 kernel; at that time there was almost no code to support this prediction, though several projects were underway. By January 2003, with the feature freeze in full swing, it seems that the resource limits were still not part of the 2.5 kernel tree. ■

INFO

The Kernel Mailing List comprises the core of Linux development activities. Traffic volumes are immense and keeping up to date with the entire scope of development is a virtually impossible task for one person. One of the few brave souls that take on this impossible task is Zack Brown.

Our regular monthly column keeps you up to date on the latest discussions and decisions, selected and summarized by Zack. Zack has been publishing a weekly digest, the Kernel Traffic Mailing List for several years now, reading just the digest is a time consuming task.

Linux Magazine now provides you with the quintessence of Linux Kernel activities straight from the horse's mouth.



■ Change PCI

It seems that /proc/pci is out of favor with Linus, and may one day be replaced by a user-space utility, lspci. This may not take place in the 2.6 time frame, but apparently Linus has been convinced that there is little reason to keep /proc/pci around, if lspci could be made to display the proper information.

Historically, the /proc/pci interface has fallen in and out of favor. The dilemma rests in the fact that /proc/pci is not absolutely essential, because the information it presents can be accessed in other ways by user-space tools; while at the same time, it is likely that those user-space tools would not be available to a system that was in the predicament of actually needing them, such as during the initial installation of the Linux system itself.

These are tricky ideas to grapple with, because it's hard to pin-point exactly when something will be needed, and whether it will be available at that time. Someone might say that Linux installers already handle PCI autodetection; while someone else might counter with the idea that embedded systems must be installed using non-standard tools. Whatever the arguments on either side, it seems that for the moment, /proc/pci is once again deprecated, and slated to be removed in favor of lspci and other fully-user-space tools. ■

■ Framebuffer woes

The Framebuffer code has still been causing problems. James Simmons and others have been working hard to straighten things out, but progress sometimes seems slow.

Even Linus Torvalds is having trouble getting things to work with the Framebuffer patches as they arrive, so getting them into the main 2.5 tree has been proving to be a problem. More fixes continue to come forward, and more drivers continue to be ported. Part of the problem appears to be the design of the Framebuffer interfaces.

Apparently, the Framebuffer code makes certain assumptions, that some video cards just don't conform to. On the other hand, it's very difficult to simply change those interfaces, because of all the user-space code that has come to rely on them over the years.

The 2.5 kernel introduces a new Framebuffer API, but it seems to be targeted at solving other problems, and will not address the deeper design issues. ■

■ Success with IDE

Andre Hedrick's IDE work appears to be going into the 2.4 tree. Such a large change to a kernel in the stable series is generally rejected, and this will be the second time in the 2.4 series that a completely new subsystem is dropped in to replace the old.

The first being the adoption of Andrea Arcangeli's Virtual Memory subsystem in the early days of 2.4. Linus took a lot of flack for that change, and shortly thereafter turned 2.4 maintenance over to Marcelo Tosatti.

Now it is Marcelo's turn to make the risky choice, but it seems that Andre's new IDE code will be met with more jubilation than Andrea's VM did.

Apparently the IDE subsystem has been such a nightmare mish-mash of horrifying hacks for so long, that Andre's cleanup just has to go in. With out this change it is feared that the system will soon prove to be unmanageable

Some of you may remember in the Summer of 2002, Marcin Dalecki had

been given maintainership over the 2.5 kernel IDE tree, and made a valiant attempt to rip out all the broken code, and bring the subsystem to a simpler, more reasonable state.

Unfortunately the politics of long-term IDE breakage proved too much for him, and he abandoned maintainership when various other developers, notably Bartlomiej Zolnierkiewicz, started their own IDE trees, instead of helping Marcin fix the main version.

At the time, Linus felt that Andre was still too difficult to work with, so Alan Cox volunteered to be the official IDE maintainer, with Andre leading the development effort itself.

Andre in turn has been trying to get his temper under control, and taking advice from folks like Al Viro about how best to organize his patches so that they will be acceptable to Linus and others. In December 2002, it seems, these efforts had started to bear tangible fruit and a sensible solution is near. ■