Zack's Kernel News

Moving forward

Linus Torvalds is barrelling toward 2.6, and says he fully intends to release 2.6.0 before the end of the 2003. To emphasize this, he has already ended the 2.5 series at 2.5.75, and has started releasing 2.6.0-test kernels.

This new development was not met with universal praise. A number of features, like support for the ARM architecture, are just not ready for 2.6, and may not make it into the early 2.6 releases at all. It's too soon to say for sure which borderline cases will make it in and which won't, but as Linus has said, these features may still make it into later 2.6 releases, without having to wait for the next unstable series.

It seems clear that Linus intends for Andrew Morton to maintain the 2.6 tree, once Linus forks off the 2.7 series. Andrew has done a tremendous amount of work on the 2.5 kernel, most notably in the ext3 journaling filesystem, and the Virtual Memory subsystem.

The road from development series to stable series is always fairly rocky, so anything can happen. Unlike the 2.1 and 2.3 series, where Linus took much longer to co-ordinate everyone's work into a suitable 2.2.0 and 2.4.0 release; this time he is being much more peremptory, insisting that developers either finish their work in time for the stable release, or wait until next time.

One interesting facet of the 2.6.0-test releases illustrates how much automation has been added to the development process in the 2.5 time frame: The BitKeeper-to-CVS gateway, and other scripts, ran into problems when the version numbering scheme went from a purely "x.y.z" format to an "x.y.z-other_text" format.

It's quite different from the migration of 2.3 to 2.4, when virtually the only bit of automation was the propagation of the 2.4.0 tar-ball to all the kernel.org mirrors. ∎

Banning orders

People who post off-topic to the linux-kernel mailing list, particularly those who start flame wars, run the risk of being banned not only from linux-kernel, but from all the hundreds of lists hosted on vger.kernel.org; posts coming from their email addresses will be filtered out, although they will still be able to read the mailing lists if they want.

David S. Miller, with support from Linus Torvalds, has sounded the battle cry, and has already banned at least one person. In response to criticism, David has said, "I know that linux-kernel is often a very un-nice place to be subscribed, and I am going to change that."

The possibility of banning unruly users first came up publicly during a recent flame-war inspired by Richard Stallman, in which Richard urged developers to try to undermine BitKeeper's hold on the kernel development process.

But it wasn't until Rick A. Hohensee posted a long description of his bash-based assembler, that David first warned, and then banned Rick from the list. Rick was perhaps a convenient test case, because his posts tend to be considered off-topic by many folks on linux-kernel, and David says he received complaints every time Rick posted.

Still, some folks accused David of being too heavy-handed with his list-admin powers, and of targeting people that he himself didn't like.

Nevertheless, it does seem as though big flame-wars may become a thing of the past on the list, or at least will move into other forums. One problem that may come up in the process of banning people, is that many developers who tend to get into flame wars, like Andre Hedrick, also produce major contributions to the kernel. Banning them from linux-kernel runs the risk of putting an end to their useful contributions as well. ∎

Crypto Kernel

Andries Brouwer, Herbert Valerio Riedel, and Fruhwirth Clemens have created the cryptoloop driver, which allows encrypted filesystems to be mounted over the loopback device. Creating an encrypted filesystem from an un-encrypted one is as easy as copying data from one device to another. Because the mechanism is based on the loopback device, it is also convenient to create small, encrypted filesystems residing on an otherwise unencrypted disk.

It is possible to encrypt a complete filesystem with the loopback device.

Encryption under Linux has always been a touchy issue, because until recently US export restrictions prevented the official kernel from supporting encryption. For this reason, a separate set of patches was maintained outside of the United States, to provide proper encryption support. This meant that encryption support in the kernel was not guaranteed or uniform and could vary amongst business users, especially if they had a USA office.

Even after the US export laws were somewhat loosened, there remained for some time the fear that they might be imposed again in the future. But finally, after some initial hesitancy, crypto patches did start to find their way into the official kernel tree, and now they are quite common. ∎

Library access

Daniel Stekloff has created libsysfs, a small library to access the features of the SysFS interface. It is intended to provide a simple way to implement SysFS support in applications, without ending up with a lot of duplicated code in each application.

Greg KH's udev replacement for the DevFS filesystem is one of the more prominent applications to make use of Daniel's library so far. Daniel also contributed heavily to the initial design of udev, on which Greg's work was based.

The libsysfs library is still quite young, but it provides generic bus, class, and device access, as represented in SysFS. It also implements a user-space applica-tion, called systool, used for listing devices by bus, by class, or by their device root.

When SysFS first arrived on the scene, it seemed that things like /proc would shortly be replaced; but this is not the case. The /proc interface has seen quite a bit of activity in recent months, and although certain interfaces are migrating to SysFS, /proc shows no sign of disap-pearing any time soon.

Nevertheless, SysFS continues to offer the hope of a clearer interface into the kernel, less sprawling and un-controlled than ioctls, /proc, or even /dev. It may be that with the advent of SysFS, those other interfaces will rediscover themselves; and identify the purposes they are best suited for, rather than continuing as disorganized catch-alls. ■

Maintaining FAT

Hirofumi Ogawa has taken over maintainership of the FAT filesystem, when the official maintainer, Gordon Chaffee, could not be found for new patch submissions. Apparently Hirofumi had been quite active in developing FATFS during the 2.5 kernel series, far more than anyone else in that time frame. Randy Dunlap actually asked Hirofumi to take on maintainership, and after further encouragement by Jeff Garzik and Christoph Hellwig, Hirofumi agreed.

This seems to be the most common way for maintainership to change hands in recent years: the official maintainer cannot be found, someone else does the majority of the work on that project, and eventually they simply take over as maintainer without any conflict and the project continues to develop at the rate everyone is happy with.

In other cases, maintainers have announced their decision to step down allowing a new maintainer to take over the project, but this is much less common.

In some cases, maintainership of a project is actively disputed even after months of parallel development, as in the case of procps. Rik van Riel and Albert Cahalan have maintained parallel procps packages for quite a long time, and neither one of them seems willing to step aside.

Fortunately, such cases are also rare, though they generate more debate than most other maintainership issues com-bined. ■

Free Cache

David Howells implemented CacheFS (not to be confused with the Apache module of the same name), a pseudo-filesystem with some very interesting features. It operates to turn any block device into a general cache of disk space, which may be formatted for use by any other filesystem, and mounted the same as any other drive.

CacheFS promises to be quite flexible, not just as a means to eek out a little more space from a tight system, or as a new way to interface with peripherals. One of the more recent holy grails of filesystem design has been the attempt to create a filesystem with version control features.

Linus Torvalds and others feel that CacheFS may be an ideal mechanism for solving the basic problems of such a goal. A version-control filesystem would retain the history of all file modifica-tions, allowing users to backtrack to earlier versions of individual files as well as entire directories.

The suggestion has even been made that such a filesystem would be ideal as one element of an eventual replacement for BitKeeper. Even so, Linus has so far been reluctant to take David's patches, just because the 2.6 series is (still only theoretically) just around the corner. But his reluctance has now slowed down the filesystem's development, and there has even been some interest in a back-port of CacheFS to the 2.4 tree.

Jeff Garzik may end up doing that work, although at the time of this writing he has not committed to it. ■

Drop the GPL

There seems to be the embryo of a move-ment to discard the GPL in favor of the Open Software License *http://www.opensource.org/licenses/osl.php*. Richard Stallman has come under increasing criticism, not only for pushing political agendas that some kernel developers disagree with, but also because a num-ber of people don't trust him to keep other aspects of his political agenda out-side of the goals of future versions of the GPL.

A number of people on the linux-ker-nel mailing list have suggested the OSL as being superior from a legal standpoint as well, and some developers have even offered to undertake the truly daunting task of tracking down everyone who has contributed to the kernel since 1991, to get them to authorize the switch. Their permission would be necessary, because each contributor retains the copyright to his or her contribution.

While a true conversion to the OSL is unlikely in the near or even medium term, a number of developers do plan to release new kernel work under the OSL. In an August discussion about a recently GPLed driver from Promise, Jeff Garzik suggested that if the driver were to be rewritten with a cleaner technical design, the opportunity should also be taken to license it under the OSL instead of the GPL.

No dissenting voice in the discussion was to be heard, and Alan Cox even pointed out that Red Hat already uses the OSL for various new projects. He also speculated that, "OSL wasn't around when the kernel began or my guess is Linus would have gone that way to avoid political baggage." ■