

Zack's Kernel News

■ FDL rejection

It's official: the GNU Free Documentation License (FDL) is not welcome in the kernel sources. The FDL was developed by the Free Software Foundation (FSF) in 2000, and was intended to address the desires of authors to make money on their work, while still delineating a proper set of essential rights and responsibilities of readers and other re-users of the text.

However, since its creation the FDL has come under some criticism. One criticism states that the FDL is not actually compatible with the GPL, and so documentation released under the FDL cannot be distributed with any GPLed projects it covers. Another is that it was just poorly thought out in other ways, and makes it difficult to make contributions to text licensed under its terms.

When some FDL-licensed documentation was noticed in the Linux kernel sources, Linus Torvalds authorized removing the files, and made it clear that other FDL-licensed files would also be unwelcome in the kernel.

The licensing situation regarding the Linux kernel is fairly complex in other ways. Each kernel developer retains the copyright to their contributions, and many contributors choose to release their work under multiple licenses.

From the initial release of Linux, Linus chose the GPL version 2, as opposed to the FSF's recommendation of version 2, "or any later version." Linus has said he didn't trust the FSF to remain true to the goals of version 2 of the GPL; and therefore chose not to release his own work under arbitrary future versions.

In addition, he released the initial version of Linux with a licensing exception allowing binary-only modules to be loaded into the kernel. Since then, it has been suggested that Linus may not have had the legal authority to make such an exception, in which case kernel modules distributed only in binary form would violate the terms of the GPL. However, no court cases have arisen to settle the various issues. ■

■ System call added

Ingo Molnar has introduced a new system call, `tgkill()`, to address an error condition in which one process may receive a signal intended for another.

This can happen when process identification (PID) numbers are used so rapidly by a given program, that the C variable used to hold the PID wraps back to zero. During such a scenario, if one process sends a signal to another, and that target process terminates, it is possible that a new process will be created with the same PID as the target process, before the signal can be delivered.

When that happens, the signal may unintentionally be delivered to the new process, causing unintended results. Ingo's new system call prevents this by not relying solely on the PID to determine a signal's target.

The `glibc` maintainer, Ulrich Drepper, has said `tgkill()` would make a good addition to `glibc`; and Linus Torvalds has also approved it. In theory, no new system calls are ever supposed to be added to the kernel, as they represent a public interface that is very difficult to change once established.

In practice, however, several new ones are added each year, and Linus has said this is to be expected. In the case of `tgkill()`, a new system call was necessary because the alternative, `pthread_kill()`, was not able to accommodate the particular error condition of PID reuse before signal delivery. ■

■ New access

Jeff Garzik has created a driver to access IDE drives through the SCSI layer. For the moment, he does not recommend using it instead of the usual IDE driver, and feels it will be most useful when Serial ATA (SATA) starts to come into its own. But he feels that the SCSI layer is most appropriate for his work, for a number of reasons.


Among these is Jens Axboe's work on the block layer in the 2.5 time-frame. For awhile, in fact, Jens' work took prece-

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.



dence over all other patches, as it involved some very invasive modifications, and Linus Torvalds wanted to get the 2.5 block layer ironed out before accepting other large changes.

As a result of that work, the SCSI mid-layer has become much sleeker, and able to handle much more of the burden's of Jeff's IDE-over-SCSI driver work. Also, Jeff feels that the structure of SATA's device and host controller implementations lend themselves to features that have already been quite fleshed out in the SCSI driver.

On a purely practical level, the SCSI driver has good interaction with other parts of the Linux kernel, such as hot-plugging support and SysFS.

By letting the existing SCSI code handle all those details, Jeff has not had to worry about them himself. On the flip side, there is still much work left to do. The driver needs wider testing before it can be considered stable.

Error handling is still quite primitive, although it does try to head off actual data corruption. ATAPI support is still quite limited, partly due to ATAPI's almost-but-not-quite conformance to the SCSI specs. For help on this driver, Jeff acknowledges Jens, James Bottomley, Alan Cox, Andre Hedrick, and tons of other developers on mailing lists and IRC. ■

■ Gateway development

Kernel developers wishing to access version-controlled sources without resorting to BitKeeper have been able to do so for some time via a BitKeeper-to-CVS gateway set up in part by Larry McVoy of the BitMover corporation. Now, an additional BitKeeper-to-Subversion gateway is also available, for people who prefer the more modern Subversion version control system over CVS.

Neither CVS nor Subversion are capable of performing the complex version control operations provided by BitKeeper, but they do preserve much of the history of kernel development in their

meta-data files, so developers and others can examine the progress of development in a patch-by-patch manner.

When Linus Torvalds first began using BitKeeper for kernel development there was a lot of dissent among some of the developers, because BitKeeper is a closed-source project. However, Linus has made it clear that it is up to the free software world to provide a viable alternative if they can.

He will not use a technically inferior tool just because it is released under a free license. While CVS and Subversion are open source software, neither of

them support the necessary features, or even the infrastructure on which to build those features.

Perhaps the most viable alternative to BitKeeper at the moment is the *arch* version control system. In theory, *arch* does support the infrastructure to allow it to compete with BitKeeper, and recently *arch* became self-hosting, meaning that developers now use *arch* as the version control system for its own sources. This is a significant milestone, and should attract interest among developers of other projects. *arch* is still very far from replacing BitKeeper, however. ■

■ Wiggle patching

Developers have a new tool to help them with patches. Neil Brown's *wiggle* tool is designed to allow users to apply a patch to a file, even when the patch and the file contain conflicting changes.

This situation can come up in any distributed software project, when people attempt to commit conflicting changes to the same file. *wiggle* was inspired by early attempts to replace BitKeeper in kernel development. BitKeeper includes graphical tools that make this kind of three-way merging very straightforward.

Any version control system wishing to replace BitKeeper would have to have

these sorts of auxiliary programs in order to be considered a viable alternative. While *wiggle* is not graphical, and is still under development, Neil feels it is a solid step in the right direction; it has already been used in back-porting features from the 2.5 tree to the 2.4 tree.

wiggle is not the first program to try to solve these problems. *tkdiff* is a graphical tool for merging two or three files, but does not seem to be actively maintained. *gtkdiff* is a similar program, likewise unmaintained.

One reason why these programs are unmaintained may be that they are

already quite usable. Another may be that a free version control system capable of making real use of them has not yet been developed. Or it is possible that they don't support the sophisticated operations needed by software developers; and would be difficult to modify to do so.

tkdiff and *gtkdiff* only operate on files, *wiggle* can also operate directly on patches that have already been created. And while the other tools operate line-by-line, *wiggle* operates word-by-word, and is thus able to find differences and matches that the others miss. ■

advertisement