ZACK'S KERNEL NEWS

Removing Old Code

A few issues ago I mentioned that JFFS was slated for removal because JFFS2 had superseded it. This has occurred! JFFS has been taken out of the main kernel tree, and Adrian Bunk has posted a patch to take out the MAINTAINERS entry as well.

David Brownell posted a patch to mark the parallel port code as unmaintained in the MAINTAINERS file because none of the four folks listed there seemed to be actively maintaining it. Jean Delvare and Randy Dunlap agreed with this assessment, and Delvare urged Andrew Morton to pick up the patch.

Joern Engel posted a patch to take the DevFS entry out of the MAINTAINERS file entirely (it had previously been listed as obsolete).

DevFS, more than any other removed feature, is one that developers have taken some great pains to stamp out completely.

The Linux kernel mailing list comprises the core of Linux development activities. Traffic volumes are immense, often reaching ten thousand messages in a given week, 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 to take on this task is Zack Brown. Our regular monthly column keeps you abreast of the latest discussions and decisions, selected and summarized by Zack. Zack has been publishing a weekly online digest, the Kernel Traffic newsletter for over five years now. Even reading Kernel Traffic alone can be a time consuming task. Linux Magazine now provides you with the quintessence of Linux Kernel activities, straight from the horse's mouth.

Specifying Code Maturity in the Config System

Robert P. J. Day has suggested adding some new maturity levels to the kbuild structure: DEPRECATED and OBSOLETE. Unfortunately, the discussion bogged down over what each of these terms actually meant. Day's idea was that "deprecated" meant the code still worked but that there was a complete alternative available, whereas "obsolete" meant the code was completely dead and unsupported. Other folks saw it in just the opposite way. Folks like Bartlomiej Zolnierkiewicz, for example, saw "deprecated" as indicating that no alternative was yet available but that the code was bad and should be replaced, whereas "obsolete" meant that a replacement was available and there was no reason to use the old code anymore.

Attempts to resolve the dispute failed instantly – there was just too much room for interpretation. On the flip side, it did seem as though everyone was in agreement that some such maturity levels would be useful; so to that extent, we might expect to see additional maturity levels added to the kernel configuration interface sometime in the future. Sam Ravnborg and Day have also discussed ways to indicate the maturity level in the option name itself at config time. Currently the only maturity level available is EXPERIMENTAL, and the only way we know we're looking at an experimental option is because whoever wrote the name of the option included a big "(EXPERIMENTAL)" at the end.

Building Under Cygwin

Deepak Saxena recently posted a small patch to allow the kernel to build under a Cygwin environment, which he said, "is unfortunately used by more people than one would think in the embedded world." The patch included various hacks, and H. Peter Anvin suggested commenting them before including the patch in the official tree so that folks touching the code in the future wouldn't mess it up.

Abortive gitweb Fork at kernel.org

Jiri Kosina recently complained that gitweb running on kernel.org seemed buggy for a while. Specifically, when switching between branches, he noticed that the shortlog always referred to the master branch.

H. Peter Anvin replied that the gitweb maintainers had not been responsive to bug reports from the kernel.org folks, and so the kernel.org folks had forked off their own version of the code.

Because they didn't have enough people or time to really maintain it themselves either, the kernel.org folks weren't able to do much better than the original maintainers.

Robert Fitzsimons pointed out that Kosina's bug had already been fixed in the upstream sources, but because the kernel.org folks were maintaining their own version, the change never made it to the kernel.org servers.

Maybe instead of a full-fledged fork that doesn't go anywhere, the kernel.org folks should do what many kernel developers do: maintain a separate patch to layer on top of each new version of the upstream gitweb.

Syslets? Threadlets!

Last month, I discussed Ingo Molnar's new Syslet subsystem, which provided a cool way to launch system calls in the background. At the time, Linus Torvalds thought the interface was too complicated and difficult for the average user. Recently, Molnar posted a much modified system, introducing the idea of *threadlets*. Threadlets are a companion to the original syslet idea, but with a far simpler interface. They are basically a way to launch arbitrary functions into the background, with the added feature that a new thread is created only if the threadlet blocks on anything; otherwise, the context remains that of the parent program. The drawback of threadlets with regard to syslets is that syslets are much faster. Speed is the price of the simpler threadlet interface.

Embedded Vendor Gets GPL Wake-Up Call

Recently, an embedded system vendor was bitten by the complex relationship between the kernel and the GPL. Three years ago, when that vendor made the decision to use Linux in their product, there didn't seem to be any issue with distributing proprietary drivers along with the kernel.

Recently, the vendor tried to upgrade to the 2.6 kernel, only to discover that to use the kernel interfaces they needed, they had to use *MODULE_LICENSE()* to release their drivers under the GPL. Otherwise, the kernel interfaces would refuse to operate. The vendor complained on the linux-kernel mailing list about this and was flamed into oblivion for doing so. However, it's clear that not all the kernel folks agree on the legitimacy of the *MODULE_LICENSE()* feature.

Without the results of a court case to clear things up, it's hard to sift through the variety of arguments to see which make sense and which don't. Regardless, almost everyone seems to agree that if a vendor isn't interested in feeding their enhancements back to the Linux community, then the Linux community isn't so interested in providing features to make that vendor's job easier, either.

Status of the Raw I/O Driver

Robert P. J. Day posted a patch to utterly remove the raw I/O driver from the kernel. It had been marked for death since 2005, and anyone wanting raw access to a device could just open that device with the *O_DIRECT* flag.

There turned out to be some controversy over this patch, though. The usual reason not to remove an interface, or at least to delay removing it, is that applications still use it. In the open source world, a deprecation period will usually inform the maintainers of all such applications, and fixes spread gradually throughout the world until at last it's safe to break the old interface.

In the case of raw I/O, one of the main groups of applications relying on it are closed source database applications. The vendors of these applications charge lots of money for upgrades. If the raw I/O driver disappeared, the only way the users of those database applications could upgrade their kernel would be to pay big fees to the database vendors for the database version that didn't rely on the raw I/O driver.

The raw I/O driver is only a few hundred lines of code, which is no great burden to leave in the kernel, so the temptation is to do nothing and not worry. Andi Kleen had a simple suggestion: take out the driver and replace it with a wrapper that just uses O_DIRECT Dave Jones and Alan Cox both liked this solution, and Jones posted a patch to take the driver off of the list of drivers marked for removal.

This pretty much satisfied Day, but Andrew Morton felt that there had to be a way to remove the interface at some point in the future. It didn't make sense to him that the kernel could be held hostage by the database vendors. Nevertheless, Kleen's wrapper suggestion was the best solution to come out of the discussion this time.