Zack's Kernel News

Requested uptake

Folks who were overjoyed to find configuration information appended compiled 2.6 kernels will be thrilled with Randy Dunlap's new addition. He got the idea to include the kernel version number and the date of compilation along with the configuration data. It appeared months after the initial decision to include configuration data in the kernel.

Often, ideas pile upon ideas, so that as soon as a feature is accepted, other developers start thinking about what they want it to do, and the number of patches increases. It's possible that the slow development of the configuration storage feature is the result of the controversy that preceded its adoption.

The first users to suggest it were flamed into silence, on the grounds that users should manage their own .config files and not bloat the kernel with unnecessary conveniences. Nevertheless, the idea stuck, cropping up every few months, and causing many seasoned kernel folks to give the standard explana-

tion over and over again: it would be unnecessary bloat. Eventually, Randy produced an 'ikconfig' implementation satisfactory to Alan Cox and some distribution vendors.

Soon several vendors were shipping with the patch included, and a version had been submitted for consideration in the 2.5 tree. Finally seven months later, in August, during the 2.6.0-test series, Linus finally accepted the patch, probably due to the vendor adoption. It was one case where the objectively 'pure' decision to avoid a feature that could be accomplished in user-space, clashed strongly with the desires of many users.

Marcelo Tosatti refused to include the patch in the official 2.4 kernel, as part of his campaign to reduce the patch intake in preparation for a new primary kernel tree, the 2.6 series. Marcelo's firmness caused many anguished cries, and he did relent on certain key patches, but not ikconfig. As a mere convenience, it did not merit bending his policy.

GFS moves back

The Global Filesystem (GFS) project has had a rough life. Initially developed by Sistina as a GPLed project to produce a globally distributable filesystem, it appeared to be on the fast track for inclusion in the official Linux kernel.

Then, Sistina decided in 2001 to rerelease the code under a newly created 'Sistina Public License'. This license allowed continued access to the source code, but required a license fee to be paid in the event of any source code redistribution. This went against some of the fundamental ideas behind open source software, and there was a huge backlash among developers.

Alan Cox claimed that Sistina had violated the license under which his own contributions had been submitted. Various other developers who'd contributed to the project, banded together to form the OpenGFS project, based on the most recent GPLed version of Sistina's code.

While OpenGFS continued its development, Sistina made attempts to capitalize on the community's earlier efforts.

Apparently none of these panned out and in 2004 their source tree was bought by Red Hat, who restored the code base to its original GPLed status.

Whether the OpenGFS project will begin to merge with the re-freed codebase is unknown, but such a reunification would not be out of character for free software development.

The early stages of development are characterized by some employees pushing an open source solution; which starts to look like a wasted gold-mine to the business folks struggling to keep their company afloat. Copyright law gives anyone the right to release their own existing work under different licenses, so not much can be done to prevent this; but at least we can welcome GFS back into the land of free software.

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.

Ouilting tool

User-Mode Linux (UML), after making heavy inroads into the official Linux kernel, has now found itself once again in a corner. The patch has grown to such a size that it would be difficult to have it accepted without a massive amount of additional work.

The standard policy, initiated by Linus Torvalds and carried forward by Alan Cox, Andrew Morton, Marcelo Tosatti and others, has been that patches should be in small, single-purpose chunks, fixing one bug or adding one feature.

Unfortunately, this is not a skill that is easy to master and the available tools do little to clear the ground. As a result, Jeff came to the conclusion that whatever system of patch management he had been using, had to be revamped.

The tool to do this is called quilt. A dangerous tool for the inexperienced, quilt has been tailor-made to Andrew's and others' development needs: maintaining a set of discrete patches against someone else's source tree.

While Andrew has said that he is willing to include Jeff's work in his -mm patches, which typically get a lot of usage among developers, and are only a short hop from the official sources; this will only tend to give prominence to the issue, generate needed bug reports and help keep that patch current against the main tree until Jeff can perform the necessary patch extractions.

Compiler changes

The GNU C Compiler (GCC) remains a central tool in the development of the Linux kernel; and much thought is given to which GCC version is the best to use with a given kernel version.

New code is constantly accepted into the kernel, to make it more portable among compilers, and Intel's compiler has only underscored this tendency. GCC remains the favorite among developers. Recently some developers have been wondering whether it would be OK to require a major GCC upgrade in the 2.7 series, to GCC 3.3 and beyond; as well as which compiler built-in directives are appropriate to use in kernel code.

The latter elicited some advice from Linus. The use of built-ins could cause problems when a user wanted to compile with an older version of GCC, or when a user compiled for an architecture in which the built-in performed worse than the code they would replace.

Furthermore, Linus feels that any actual advantages of built-ins will be

minimal even in the best case. He has therefore advocated the use of built-ins in only two cases: either if the built-in is old enough to have been thoroughly debugged and made available on all (or almost all) architectures; and if using the built-in results in a clear and measurable improvement.

It's possible that this reluctance is based on earlier disputes between kernel developers and the GCC team. More than once, kernel folk have refused to upgrade their GCC, when they felt the compiler was moving in the wrong direction.

GCC folks have strong ideas about the way things should be, and they feel an obligation to all of the projects that rely on GCC; not just the kernel. Also, they have enjoyed the advantage of not being dependent on Linux, while for much of its existence Linux was entirely dependent on GCC to perform its compilation.

While outright animosity has been on the ebb in recent years, both groups do not refrain from harsh criticism.

■ No stopping suspend

The swsusp project has been trying to support SMP for some time, and Pavel Machek thinks he and the other folks working on the project have finally got it right. The swsusp project, or Software Suspend, is an attempt to enable a running system to shut down and then later resume with its state unchanged.

One of the primary obstacles to success is the fact that it is not always possible to know the state of hardware on a system prior to shutdown; nor is it possible to guarantee the state of hardware during or after resumption of a suspended system. Attempts to solve, or sanely deal with this have made the whole issue controversial.

It should be remembered that Linux developers are highly motivated to continue to improve swsusp, in part because commercial operating systems have offered similar features for quite some time, and it always stings a little to hear of something a commercial OS can do that Linux cannot.



OUR COMMITMENTS **:

• SATISFACTION OR REFUND • NO SET-UP FEES • NO HIDDEN FEES • 7/7 TECHNICAL SUPPORT • UPGRADE FREE • ADMINISTRATION 100 % ONLINE • PROACTIVE MONITORING 24 X 7 • 99,9 % AVAILABILITY • REDUNDANT NETWORK • GUARANTEED BAND WIDTH

www.amenworld.com

0870 444 6778

