Zack's Kernel News

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.

License issues

Pat Gefre and other folks from SGI have released a driver for the Altix serial console. However, there may be problems integrating it into the kernel, because of the possibility that SGI is imposing additional restrictions on its code, beyond those imposed by the GPL.

The Free Software Foundation has apparently approved SGI's license as being fully in compliance with the GPL, but some kernel developers are concerned that SGI is explicitly reserving the license to any patents involved in the code they submit. This could have the effect of preventing SGI's contributions from being used in other GPLed projects, or even in a fork of the Linux kernel itself.

The question for kernel developers may not be one of whether SGI has violated the GPL, but whether they want to include code in the kernel that is subject to patent restriction. The patent issue (and the GPL's lack of a firm stance on it) is one reason why many kernel developers favor a migration from the GPL to the OSL, which apparently takes these issues more into consideration. ■

The right to run free

Makan Pourzandi and other folks on the DSI (Distributed Security Infrastructure) development team have created the digsig tool, which verifies the digital signature of a binary before running it.

A signature embedded in a binary may be checked via digsig before execution, and run or not run based on the result. digsig depends on hooks provided by the Linux Security Module, and may be run independently of the rest of DSI's kernel modifications. This could be useful for protecting against worms, or for implementing some form of Digital Rights Management.

Linus Torvalds has already said that DRM patches are perfectly acceptable in the kernel, so long as they take the form of general-purpose enhancements, rather then being targeted to particular uses and situations. ■

Higher profile

Juan Villacis and other folks from Intel have produced some enhanced kernel event-notification code, which they would like to see go into the 2.6 kernel. Their code is intended to enable more powerful code profiling, by giving profiling hooks not only at the end of a task's existence, but at its creation as well.

In theory, all the various profiling packages out there would benefit from the patches. The question remains as to whether Intel's code works best on its own, or whether it should be layered above one of the existing profiling tools, such as oprofile. Because so many different folks are involved in writing profiling tools, it's important to avoid duplication of effort.

There was also the initial suspicion (since mollified) that Intel's code might provide a mechanism for proprietary software vendors to bypass the GPL, essentially creating fully integrated offshoots of the Linux kernel, that were not bound by the terms of the GPL.

Andrew Morton, tipped to be the 2.6 maintainer when it comes out, voiced this objection; but it was pointed out that Intel's code was only adding functionality that existed in similar form elsewhere, and could not be used in the ways Andrew feared. ■

Check the plug first

Bryan O'Sullivan has created Netplug, a daemon that automatically brings a network on- and off-line, depending on whether the networking cables are plugged in. Essentially, it is network hotplugging, and should be quite useful for users with laptops, as well as systems in a clustering environment.

This is not the first time such a task has been attempted. ifplugd has been around for awhile, and is apparently quite usable and actively maintained, but did not use the standard link-detection mechanism found in Netplug. It is possible the two projects will merge their best features in the future. ■

On your marks, Get set, Compile!

In the old days the Linux kernel could only be compiled with GCC, but now that Intel's ecc compiler has reached the point where it can compile the kernel as well, inevitably conflicts and incompatibilities have arisen.

Every once in awhile someone posts a patch specifically to enhance compilation with one compiler or the other. Most recently, some patches surfaced to compensate for the fact that ecc didn't support inline assembly.

Now, many developers feel that inline assembly should be removed from the kernel wherever possible, regardless of which compiler is used. At the same time, there is a case to be made that it is the responsibility of the compiler to support the features required to compile the kernel. After some arguing back and forth, and some suggestions from Linus Torvalds, it was decided that compiler-specific code should all be gathered together, much as architecture-specific code is gathered in the arch/ directory.

This would theoretically allow other compilers to tackle kernel compilation, modifying the kernel to compensate for their limitations, without inconveniencing any other compilers. As soon as the decision was made to isolate compiler-specific code, Sam Ravnborg and David Mosberger jumped on it, submitting patches to get the ball rolling. ■

Slabtop

Chris Rivera and Robert Love have created slabtop, a top-like utility that displays slab information in real time. In the kernel, a sophisticated memory caching mechanism is implemented, which allows memory to be reused, without having to allocate and deallocate it all the time.

Memory caching maintains the allocated memory in a pool that can be made available on request. However, sometimes the system really does need free memory to be available, and so memory currently in the cache needs to be deallocated. The slab layer handles this.

A slab cache is very useful when large data structures will be often created and destroyed, because it saves several steps in the process of organizing and allocating memory. The slabtop tool displays a running total of basic statistics for system-wide slab activity, as well as continuously tracking individual cache data on a slab-by-slab basis. Slabtop looks similar to the top utility

Changing history

An interesting cultural debate has cropped up. Should it be possible for Linus (or anyone) to modify BitKeeper commit messages after a patch has already been incorporated into the tree? Linus Torvalds thinks so, and does it regularly, to improve readability and to indicate where a patch claimed to do something that it was later found not to do.

The debate began when Albert Cahalan noticed that a changelog entry was wrong, and asked if it could be fixed. Many people objected to this on a variety of grounds. For one thing, they said, as changelog entries were not themselves under version control, modifying them was in effect a violation of the versioning system provided by BitKeeper (or any other system).

The true history of the project would be lost, or at least cast in doubt. It was argued that legal hassles could ensue, if ever a time came when a particular changelog entry would prove pivotal to a court case. Lately, with the SCO hassles, this has not been an inconceivable situation.

Visible headers

In preparation for 2.6.0, Matthew Wilcox has been trying to separate kernel headers into those that should be visible from user space and those that shouldn't.

One perennial problem in Linux has been users including kernel headers in their user-space code; which can cause no end of headaches for code maintenance and dependencies.

If user tools begin depending more and more on kernel headers, this could one day put pressure on kernel developers to avoid breaking user tools by leaving the kernel headers unchanged. This would restrict the ability of kernel folks to improve and enhance the kernel.

Matthew aims to put an end to that problem. At the same time, Erik Andersen and David Woodhouse have been inspired to clean up various naming and type conventions; the danger being that if they go too far, they may end up violating various standards.

Ordinarily the Linux kernel does not shy away from violating standards that make no sense, but even Linux has to consider the impact on users and on user-space code.