

Zack's Kernel News

■ Doorway to BitKeeper

Pavel Machek has set up a CVS gateway to BitKeeper, so developers who want to use only free software, can use CVS to communicate with the BitKeeper trees maintained by Linus and others.

Ever since Linus began using BitKeeper to organize development, the kernel developers have been split into two camps. One camp feels that BitKeeper solves a lot of problems and is a good thing to use, especially as there is no free alternative; while the other camp feels that Linus, as spokesman for the entire community, should not compromise the ethics of free software by giving such a central role in kernel development to a commercial product.

The most visible advantage to BitKeeper is that each new kernel release is now accompanied by a complete description of the patches that went into it. But many people feel that this and other advantages are outweighed by the fact that BitKeeper is a commercial, closed source product.

Various alternatives to BitKeeper have sprung up recently, but none of them have achieved the technical maturity of BitKeeper, and Larry McVoy (BitKeeper owner) predicts that it will take years to develop a free alternative to BitKeeper. So Linus and a number of other kernel developers continue to use it. ■

■ Joining up Filesystems

Filesystem capabilities are making progress. There's been partial support since 2.2, with several individuals and groups working on the problem ever since. Now it looks as though complete support may arrive within the 2.5 time frame. The 2.5 Virtual Filesystem (VFS) has supported extended attributes (EAs) since 2.5.3, and plan to implement POSIX capabilities within the EA framework.

However, the Linux Security Modules (LSM) project has been coming at the problem from the opposite direction, implementing capability support, without

■ Quick freeze

A 2.5 feature freeze is planned for October 2002, as decided at the recently held Linux Kernel Summit. A code freeze will follow, in which only bugfixes will be accepted into the kernel; and finally, 2.6 will be released, amid joy and jubilation around the world. That is the plan. The reality, however, will almost certainly prove somewhat different.

Earlier transitions from unstable series to stable releases have all taken much longer than anyone expected, and this has been recognized as a problem for years, not just in kernel development, but in many other large open source projects as well. In open source, there are many developers working at all times to add features, rewrite various existing portions of the kernel, port the system to other architectures, and so on.

As long as the development series is in full swing all is well. These developers may work and work at their own pace, concluding their work when the time is right. But the 2.6 kernel cannot be released until all the various tendrils of development have been brought together, at least somewhat, or the system would not work at all.

It's quite common for kernels in the development series to be broken and not even to compile successfully. This is because work in one area may be at a

yet handling the link between the EA framework and capabilities. It seems that all that remains is to meet in the middle, which does not seem such a long way off. POSIX capabilities allow root privileges to be split up into atomic privileges that may be granted or withheld individually.

A given program may run with the capability to delete a file on the system, but not to modify it. Extended attributes are a general purpose method of storing metadata within an inode on disk. Each attribute is composed of a name and a corresponding value, stored with the file.

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 decisions and discussions, 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.



particularly invasive stage, while a new release is required in order for other developers to continue merging their work. Before the transition to a stable series, however, all the developers must bring their portions of the code to roughly equal status, as complete and as stable as they can get them.

Naturally there is always a big push to get just one more feature in before the deadline, and in earlier years Linus would often make such exceptions, which then needed time to stabilize, during which other people would protest the exclusion of their own patches. If 2.5 does successfully freeze in October, it will be the shortest development cycle on record, and will indicate a shift in the way it has been handled. ■

As with many new Linux features, capabilities, extended attributes and Access Control Lists have been extremely controversial at times. The developers tend to take a unique approach to all aspects of system design. It is not unheard of them to reject the accepted standards, if they feel a better solution is available. As a result, the question of whether to add a particular feature like capabilities often boils down to details of implementation and behaviour that may not have been envisioned by its original designers and developers. ■

■ Time for a change

Ever since Ingo Molnar wrote his fast new process scheduler, there has been a tremendous push to see it included in the 2.4 kernel. But for six months the maintainers have resisted including it.

A number of Linux vendors have included the patch in their distributions with no problems, so that most Linux users in the world have probably been using the patch for some time, but among the kernel developers there is reluctance to include such an invasive change into the 2.4 series, which is supposed to be kept as stable as possible.

Any patch to modify a fundamental component as the scheduler, would make the kernel less dependable, because it would be less well tested. Ingo hopes is that the patch will receive more testing, and be shown to be truly stable, before being included in the stable kernel series.

Some developers feel that the patch should wait for the 2.6 series. These developers point out that the default scheduler currently in use in the 2.4 series is perfectly usable, and doesn't need to be replaced.

It is impossible to think about this issue without recalling Linus' decision to replace the entire Virtual Memory subsystem early in the 2.4 series. This was a very invasive change on the order of replacing the scheduler, and was met with harsh criticism and much bitterness.

In addition, the VM subsystem at that time still had many problems, and was improving only very slowly. In the event, it turned out that Linus' decision to replace it, led to a more robust system, though many developers felt he should not have taken such a big risk. ■

■ 2.0 marches on

Someone recently suggested dropping support for the old 2.0 kernels. David Weinehall, the 2.0 maintainer, said he would continue to patch 2.0 bugs as long as people continued to send fixes to him. He said, and Alan Cox (2.2 maintainer) agreed, that maintaining these kernels did not drain development effort from more current projects, primarily because there was so little required to maintain them. David predicted, that so little work needed to be done on 2.0, that he would

■ Temporary unstable fix

The 2.5 IDE disk code is being entirely rewritten for 2.6, and is currently in a broken state, which has been causing delays in other areas of development.

Some developers attempting to test their own work on recent 2.5 kernels have been unable to do so, because IDE support has been removed during the extensive changes. Recently it was reported that system lockups and even data corruption could result from testing the current 2.5 IDE code.

While this is not uncommon for a development series, it has caused some frustration among various developers, and recently inspired Jens Axboe to port the 2.4 IDE code up to 2.5 and maintain it as a separate patch. He did this in order to be able to test his own projects, but thought other folks might find it useful.

In fact, many people were overjoyed by this development. Some developers had been too frightened even to try any 2.5 kernels, but with IDE temporarily patched up, they felt they could begin to reach tentatively into doing 2.5 work.

Jens was careful to add in his announcement, that the IDE maintainer was doing a very good job, and that Jens' patch was simply a temporary expedient until the real IDE code stabilized. It is this tactful acknowledgement that probably prevented an angry flame war.

The IDE rewrite has been controversial, because it has had to get worse before it could get better. Most large rewrites have either not entailed long periods of breakage, or else have involved less central systems, whose breakage would not inconvenience too many developers involved in doing other work. ■

probably only release 2 or 3 additional versions, and will almost certainly stop with 2.0.42.

Amusingly, Mikulas Patocka recently refused to take over as maintainer of the 0.01 tree. Last September, while playing around on the earliest version, Mikulas discovered a bug and posted a fix on linux-kernel. A lot of folks' eyes popped out of their heads over that one, and Linus offered to let Mikulas be the official maintainer of that tree.

■ Stable Detection

Hardware detection is reaching stasis. In the desire for a fully developed plug-and-play system, the question occasionally comes up, of how to automatically detect all hardware currently installed, and how to detect hardware that is hot-plugged into and out of a running system.

Current kernel policy is to detect all hardware that it is possible to detect, but not to make assumptions about the ways that hardware will be used. For instance, it would be a security risk for the kernel to automatically mount all filesystems it detected at bootup. The decision of when to mount the filesystem is left to the administrator, even though most Linux systems mount their filesystems at bootup, it is controlled by user-level configuration, not by the kernel.

The situation is made more complex by the fact that Linux runs on a great variety of systems, which don't all support the same kinds of hardware detection. Some systems, such as s390, s390x, x86 and ia64, are now able to hot-plug CPUs in and out of the system at will, while others show no promise for such a thing.

For a long time, developers despaired of ever being able to hot-plug regular PCI cards, until Compaq demonstrated a Linux system capable of this in January of 2001. Within a couple months, patches for this were in the mainstream kernel sources. But it remains dangerous to hot-plug certain pieces of hardware. On some hardware, plugging a mouse or keyboard into a running system may break those components. There is nothing the operating system can do about those situations, because the problem occurs at a more fundamental level. ■

Then it was Mikulas' turn to have his eyes pop out of his head, and sadly, he refused to honor. Maintainership has often been delegated based on interest. Alan became 2.2 maintainer primarily because he insisted on producing patches for it. David became 2.0 maintainer because he objected when Alan decided to stop maintaining 2.0 himself. As far as I know, Marcelo Tossatti (2.4 maintainer) is the only person to actually go through a selection process. ■