conference reports

OUR THANKS TO THE SUMMARIZERS:

2005 LINUX KERNEL DEVELOPERS SUMMIT
Jonathan Corbet

WITMEMO '05
David Blinn
Minkyong Kim
Irfan Sheriff

EESR '05
Ahmad T. Al-Hammouri
Apratim Purakayastha
Himanshu Raj
J. Sairamesh

MOBISYS '05
Mike Blackstock
David Johnson
Neil McCurdy
Xiaoqiao (George) Meng
Himanshu Raj
Ram Kumar Rengaswamy
Ya-Yunn Su
Denitsa Tilkidjieva
Hongwei Zhang

HOTOS X
Prashanth Bungale
Rik Farrow
Alexandra Fedorova
Nikolaos Michalakis
Steve VanDeBogart
Steve Zhang

VEE '05
Xing Fang
Long Fei
Shuo Yang

SRUTI '05
Jayanthkumar Kannan
Balachander Krishnamurthy
Lakshminarayanan Subramanian

2005 Linux Kernel Developers Summit

Jonathan Corbet is a co-founder of LWN.net and the author of its kernel content. He is the lead author of *Linux Device Drivers,* 3rd edition, published by O'Reilly. For the last four years, Jonathan has been on the planning committee for the Kernel Summit.

In the young and fast-moving Linux community, anything that has happened for four years in a row can be called "traditional." Thus, the 2005 Linux Kernel Developers Summit, held on July 18 and 19 (immediately prior to the Ottawa Linux Symposium), is by now a traditional event. For two days each year, this invitation-only crowd of around 70 core kernel developers gather to talk about where kernel development should go over the next year. Few important development decisions were made at the 2005 Summit, but it was an opportunity for developers to catch up on what is happening in areas outside their particular expertise and, of course, to pursue topics of interest in the hallway and pub meetings.

The 2005 Summit opened with a panel of processor architects. This panel has, over the years, served as a forum where manufacturers could share some of their plans and hear about any concerns the kernel developers have. Two themes stood out this year: power management and virtualization. Manufacturers need to reduce the power demands of their chips lest future systems be required to be equipped with cryogenic cooling units; they would like to have help from the kernel developers in designing algorithms (scheduling in particular) that can help with power management. The developers, in return, would like a reliable way to ask the processor what its current power consumption is so that power-related changes can be benchmarked. There is quite a bit of hype around virtualization—running guest operating systems on top of software-implemented virtual machines—and the processor manufacturers are responding by adding better virtualization support to their CPUs.

A session on I/O busses mostly concerned technical details on the best interfaces for DMA operations and dealing with memory-challenged devices and systems. The kernel contains several independent mechanisms for setting up and executing scatter/gather I/O; it was agreed that it might be nice to unify these subsystems at some point, but nobody seems in any real hurry to do the work. There was discussion of a new memory allocation interface that would help drivers work around the memory addressing limitations found in a discouraging number of devices; a patch is expected soon.

The virtual memory management session showed that nobody is itching to make major changes to how the VM subsystem works—in the near future, at least. Instead, attention is currently focused on dealing with memory fragmentation and memory pressure. The most likely short-term solution to fragmentation (where it gets hard for the kernel to allocate multiple, contiguous pages) is a new allocation scheme that would segregate user-space pages (which are easily moved) from kernel memory allocations (which are not). Linux still does not behave as well as anybody would like when memory gets seriously tight; the issue here seems to be finding a good way to throttle memory-intensive processes without creating performance problems.

A brief session discussed some security-related patches that could be merged soon. These include better kernel API checks (finding double-free errors, for example), more address space randomization, making use of recent gcc features, and tightening access to various files in /proc and /dev/mem.

A session dedicated to virtualization had little to say; most of that work is in user space at this point.

There is interest in merging the Xen patches sometime soon. Those patches have been significantly reworked (Xen used to add itself as an entirely new architecture, but that did not go over well with the kernel developers) and should find their way into the kernel before too long.

The final session on Monday was dedicated to the virtual filesystem layer. Some potentially contentious issues (such as the merging of Reiser4 or FUSE) were avoided entirely; instead, the discussion centered on the increasing complexity of the core VFS code. In particular, the mixture of direct and buffered I/O has created a difficult mess that somebody will eventually have to clean out.

Tuesday began with a panel that addressed the frustrations faced by hardware manufacturers who wish to work with the kernel development process. The corporate way of doing things, involving fixed schedules, lengthy internal quality assurance work, and control over the code, does not mix well with the Linux process. Getting code into the kernel is easier if that code is posted at a very early stage so that show-stopper problems can be identified and fixed. But hardware companies would rather just produce a fully functional, tested, and certified driver at the end. That approach can get them sent back to the drawing boards with fundamental problems to fix. The vendors also complained about the constantly changing kernel API, which gives them long-term support problems.

The networking developers had held a summit of their own the week before the Kernel Summit; the outcomes were summarized for the crowd. A great deal is happening in the network community, including the reworking of much old code, the de-bloating of the core sk_buff structure (which represents a packet in the kernel), better cryptographic and security support, and more. We may even see support for hardware TCP offload engines, something that has been resisted by the networking developers for years.

From networking, the discussion turned toward the increasing convergence of the networking and storage subsystems. Storage-area networks, iSCSI, and so on are making networking a crucial part of the block I/O subsystem. This convergence can cause problems when memory gets tight; the block layer needs to write out pages to free memory, but a network-based storage layer must allocate memory to accomplish those writes. There are things that can be done to address this problem, but Linus Torvalds also wants to push back on the manufacturers of these systems. Rather than go through all this trouble to make network-based storage work, wouldn't it be better to just install a local disk? That said, there are real reasons behind these technologies, and Linux will find a way to support them properly.

A brief session on clusters showed that there was not a whole lot to concern the kernel developers; once again, most of the work is now in user space. There will be moves to merge a couple of cluster file systems soon (RedHat's GFS and Oracle's OCFS2); it seems that the two might have agreed to use the same distributed lock manager.

The session on RAS tools was mostly a celebration of the merging of the kexec and kdump patches, which should bring reliable crash dump capability to the mainline kernel. There are still quite a few loose ends to tie down.

Real-time capability for the Linux kernel has been the subject of a great deal of intense discussion over the last year. Most of that intensity failed to show up at the Kernel Summit session dedicated to the topic, though. There was some talk of how the various ways of providing real-time response could be judged, but no time to actually apply those criteria. So real-time can be expected to continue to heat up the mailing lists for a while yet.

The Desktop Developers Conference was happening at the same time as the Kernel Summit; a few delegates came over to give the kernel developers an update. The core of the discussion consisted of grungy details on how to rationalize Linux support for graphics cards. These cards are complex devices, often with secret interfaces. In the past, there has been a great deal of confusion as to whether these cards should be controlled by the kernel or by the X server in user space. These issues are slowly being worked out, and better graphics support should be coming to a screen near you shortly.

The kernel developers heard a report from the power management summit, held two days earlier. Much work remains to be done in the power management area. There are currently two software suspend implementations, neither of which is as solid as its users would like. It was agreed that the external "suspend2" patches would be posted and considered for merging into the mainline. Video adapters are a constant challenge in making suspend work; they are supposed to be reinitialized by the operating system on resume, but the manufacturers will not tell the Linux developers how to do that. So, instead, pressure is now being put on the BIOS vendors to provide that reinitialization support in the firmware.

The final session, traditionally, is devoted to the kernel development process and the ongoing desire to extract hard deadlines from Linus Torvalds. Deadlines were less of an issue this year; instead, the developers were concerned with improving the quality of kernel releases; recent kernels are seen by many as containing too many bugs. Two

reasons were identified for this: kernel developers are waiting too long into the release cycle to merge their changes (thus missing out on weeks of testing time), and bugs, even when identified, are not being fixed. An attempt will be made to address the first problem by requiring that new features be merged into the kernel within the first couple of weeks of the cycle. After that, a feature freeze of sorts will be imposed, and only fixes will be merged. Getting developers to actually fix bugs can be a bigger challenge when there is no boss to order them to fix things.

Overall, the 2005 Summit was seen as a successful gathering. Some developers have noted that, over time, the summit is moving away from a forum where issues are debated and decided and is becoming instead a two-day status report. Given that the kernel has grown to a point where nobody can really understand every part of it, such a status report can be important. But if the summit is not a place where decisions are made, some of the developers may stop coming. So there may be changes made in the future to spice things up a bit.

For more detailed reporting from the summit sessions, please see http://lwn.net/Articles/KernelSummit2005/.