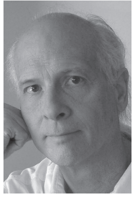
RIK FARROW

musings



Rik Farrow provides UNIX and Internet security consulting and training. He is the author of *UNIX System Security* and *System Administrator's Guide to System V* and editor of the SAGE Short Topics in System Administration series.

rik@usenix.org

FOR THE PAST COUPLE OF MONTHS, I have become absorbed in operating systems. My quest began with some consulting regarding security features of the current Linux kernel, including SELinux, then plunged even deeper during the HotOS workshop. I do not pretend to be a kernel hacker, but I am very interested in what goes on with the design and implementation of operating system software.

Like some others, I had wondered what had happened to FreeBSD 4's stellar performance when FreeBSD 5 appeared. Instead of getting faster, FreeBSD was slower. If I had bothered digging deeper, I would have learned that this was the result of far-reaching changes in the FreeBSD kernel. Michael W. Lucas explains these changes in his article, which in turn is based on a talk given by Robert Watson about modifying the kernel to support SMP (symmetric multiprocessing). And perhaps by the time you read this column, FreeBSD 6 will have appeared, ready to utilize the new multiprocessor cores that are popping up.

Lucas explains just why the transition from single-threaded to multi-threaded kernel takes so long and is so hard to do right. I first understood the importance of the Big Giant Lock when I was reviewing an early multiprocessing server that used SPARC processors. I ran a simple benchmark that spawned additional processes, each of which ran an integer-intensive program. I tried my benchmark with one processor, then two, three, and, finally, four processors enabled, and the results astounded me (at the time). Adding processors does not in itself linearly improve performance. Enabling the fourth processor barely added a 15% improvement to the results. The Big Giant Lock ensures that only one process (or interrupt handler) can run in kernel space at a time, which devastates performance.

The HotOS workshop (see the summaries in this issue) brought different surprises. I enjoyed the workshop immensely, as much for the free time spent with attendees as for the talks. The lunchtime discussions have sparked one article already, in which Marc Fiuczynski makes a fervent plea for better methods for patching Linux kernels.

You will also find a discussion of the Linux Kernel Developers Summit by Jonathan Corbet. Corbet, who has been summarizing the summits for years, has provided a short overview for *;login:* readers. And Peter Galvin explains Solaris 10 containers. Note that Sun bit the SMP revision bullet years ago, making it a leader in SMP OSes today. The concept of containers adds a powerful twist to Solaris, a useful VM architecture unlike others you may know about.

On the security front, David Malone discusses security features of FreeBSD. As I read this, I found myself wishing that some of these appeared in Linux as well. But BSD-envy is nothing new. Srikanth Kandula explains Kill-Bots, based on a paper he presented at NSDI (see the summaries in the August issue of *;login:*).

You might note that instead of my usual musings, I have acted much more like an editor this time around. I apologize, but this issue is so packed with summaries and articles, I really didn't have the space for me. You can expect that my column will appear much as it has in the past in the December issue of *;login:*, with its focus on security.

Again, I do appreciate your feedback about the changes appearing in *;login:*. You can send me email at login@usenix.org, along with article proposals, letters to the editor, complaints, and praise. If you have books you want to review (or think should be reviewed), try the new bookreviews@usenix.org alias. You can find a more structured approach to making suggestions about *;login:* at https://db.usenix.org/cgi-bin/loginpolls/oct05login/survey.cgi.

To be honest, I am quite happy to be able to approach some of the world's brightest minds and ask them to write articles about those issues I consider most crucial for the advancement of computing systems. But you do need to make sure I don't stray too far.

USENIX FAST 20 05 4th USENIX CONFERENCE ON FILE AND STORAGE TECHNOLOGIES
DEC. 13-16 / 2005 / HOLIDAY INN GOLDEN GATEWAY / SAN FRANCISCO / CA /USA

SAVE THE DATE!
FAST '05
4th USENIX Conference on File and Storage Technologies
December 13–16, 2005, San Francisco, CA

Join us in San Francisco, CA, December 13–16, 2005, for the latest in file and storage technologies. The 4th USENIX Conference on File and Storage Technologies (FAST '05) brings together storage system researchers and practitioners to explore new directions in the design, implementation, evaluation, and deployment of storage systems. Meet with premier storage system researchers and practitioners for a day of advanced tutorials and 2.5 days of ground-breaking file and storage information!