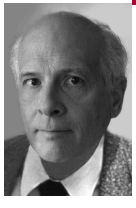
RIK FARROW

musings



rik@usenix.org

Information technology, IT—the branch of engineering that deals with the use of computers and telecommunications to retrieve and store and transmit information.

—wordnet.princeton.edu

WHEN I FIRST READ ONE OF THIS issue's articles, I found myself suggesting to the author that he not use the term IT. I wrote back to him, saying that many of the readers of *;login:* are system administrators and that they don't consider themselves part of IT.

Having said that, I wondered how true that is. I also pondered over my own aversion to being considered part of IT. The definition of IT appears benign (see sidebar). It even appears to match, in very broad terms, what it is that system, network, and security professionals do.

The Priesthood

My earliest vision of IT came with my first visit to a computer room, sometime about 1961. I had volunteered to work with some "computer" people (no "IT" people back then) and write a program in assembly language that would run on a mainframe. The mainframe itself, in Rockville, MD, was tremendously impressive, with its many large, humming cabinets, the washing machine–sized disks with the intriguingly labeled (in red) "Write Protect" push buttons, and the CPU itself. I liked the flashing lights that showed memory addresses and data values, and it was sort of cool when the IT guy got the mainframe to play a song through the console speaker (poorly). But I was just as impressed by the spectacular air conditioning system, as the air inside the computer room was marvelously filtered, cool, and dry.

In 1969, I spent a summer working as an intern for GE in Bethesda, MD. I was the "software librarian," and the only really cool part of my job is that I got total use of the mainframe, during lunchtime, about once a week. I learned to boot the mainframe, starting with binary cards, then a larger deck of Hollerith-encoded punch cards, then bootstrapped the terminal concentrator, and finally got the tape drive, which contained the master copy of the OS and utilities, going. Once the system was up, I installed patches, from other nine-track tapes or from paper tapes, then dumped the patches to new nine-track tapes for shipment to the other 18 mainframes like this one around the world.

I took every CS course I could at the University of Maryland, which didn't even have a CS degree in those days. Using punch cards, and submitting jobs that you would not see the results of for at least 4 hours, and sometimes 20, really helped with

typing discipline, as a single typo was tremendously painful. You would hand your card deck over to the priesthood who managed the CS department's mainframe, and they would load your deck into the card reader and then write it to nine-track tapes. The tapes, containing card images, got loaded on the mainframe, and the jobs would run one at a time. I later learned that there was some time-sharing going on, but most students were forced to run batch jobs. (With some 25,000 students and a single computer, we were fortunate that most were not doing anything with the computer.)

None of these experiences should have left me with negative feelings toward being part of IT. I did not work with computers my first several years out of college, because most jobs went to people with experience, and the Vietnam War had provided large groups of people with real experience, seriously impacting the market for those, like myself, who got college degrees instead of traumatizing experiences.

When I did get back into computers, it was because the microprocessor had started appearing in small systems. During the intervening years, DEC had been producing minicomputers, smaller, cheaper, and much less complex systems than their mainframe relatives, but even these were scarce. Microcomputers promised to change all that, making real computers available to everyone. I could see this clearly in 1978, and I quickly became a part of the early PC boom in the Bay Area.

Perhaps my aversion to IT came later, when I was working as a sysadmin consultant. I would have experiences where I would learn about some problem an IT staff was having and offer to take a look at it. For example, one company needed to migrate its database from a homebrew version to Informix. Both ran on SunOS, but the dump format of the homebrew version was incompatible with the import format of Informix. I talked to the IT guy who was struggling with the issue, figured out how the two formats differed, and wrote a short shell script that managed the conversion. All this took perhaps an hour.

The IT guy was amazed, saying that he had expected that it would take weeks of work for the IT department to do what I had done in an hour. I was perplexed. It really wasn't that hard, I thought, so what gives? Then I learned that such occurrences were the norm, and that getting anything done through the IT department was a bureaucratic nightmare that could take weeks or months.

Perhaps you can tell me if your own perceptions of IT are at all like mine. Perhaps you like being part of IT. I really don't know. I would be happy to release old, misbegotten perceptions, as the world moves much faster now, and I imagine that IT projects done by foot-dragging office drudges are now things of the past.

Lineup

We start out this issue with an article about Plan 9. Although Plan 9 may have faded from people's memories, the poster a group of HPC scientists had during USENIX Annual Tech generated considerable interest, so I thought it might be time to revisit Plan 9. Andrey Mirtchovski and Latchesar Ionkov don't want us to forget the many useful, yet simple, ideas that are part of Plan 9. In the case of Linux, the Plan 9 communications protocol, p9, has actually been ported and is still supported today. Plan 9 has much to offer, especially in the world of HPC, but also to students who need to be exposed to alternatives to the Linux OS design.

Cat Okita writes about Identity 2.0. I had heard Cat speak at LISA'06 and asked her to update her presentation for this issue. Identity goes beyond sysadmin, as it is something that affects every citizen of First World countries, and whose impact continues to expand.

Lex Muentz wants you to get ready for the possibility of litigation, as Federal Rules of Civil Procedure are morphing. These rules govern the handling of material required to be handed over during civil suits in the U.S. Lex clearly explains where the legal profession believes the landscape is changing. His advice on how to cope with electronic evidence will not only save you time and money but may help your organization avoid stiff fines that could be levied on you because you weren't prepared.

Octave Orgeron presents his second article about LDoms, the new Solaris virtualization capability. Logical Domains run above a hypervisor on some newer Sun systems, which does limit the number of people who will immediately be using these systems. But the techniques used in LDoms will, I believe, start appearing on other systems, as we already have other hypervisors, from VMWare and Xen. Octave's article explains how Sun goes about installing support, including firmware and software, for the LDoms system.

Hemant Sengar discusses some unsung dangers in VoIP protocols. His concern centers on the protocols used for setting up calls, protocols that can easily be abused for DDoS attacks and for DoS attacks with amplification.

David Blank-Edelman shows how you can draw pretty pictures, well, uh, diagrams, using Perl, and he highlights some easy-to-use libraries. David Josephsen describes how you can pry out information about what goes on within Java Virtual Machines. Debugging performance issues within JVMs requires access within the JVM. David explains tools that you can use to do this, as well as how to export performance data to your monitoring systems.

We have book reviews, of course, and Robert Ferrell explaining to us why we need to get rid of DNS. Yes, you heard me, but I will let Robert explain. Nick Stoughton has written an excellent article about the progress toward a new C standard.

Dan Wallach tells the story of how the USENIX/ACCURATE Electronic Voting TechnologyWorkshop came into existence. If you have a new field of interest that you think might fit the workshop model, don't miss Dan's short article in this issue's "USENIX Notes."

We have two sets of summaries in this issue: the 2007 USENIX Annual Technical Conference and the Linux Symposium 2007.

It seems to me that my early experiences with IT did not create the antipathy I grew to feel later. But perhaps things have changed. What's your take on it?