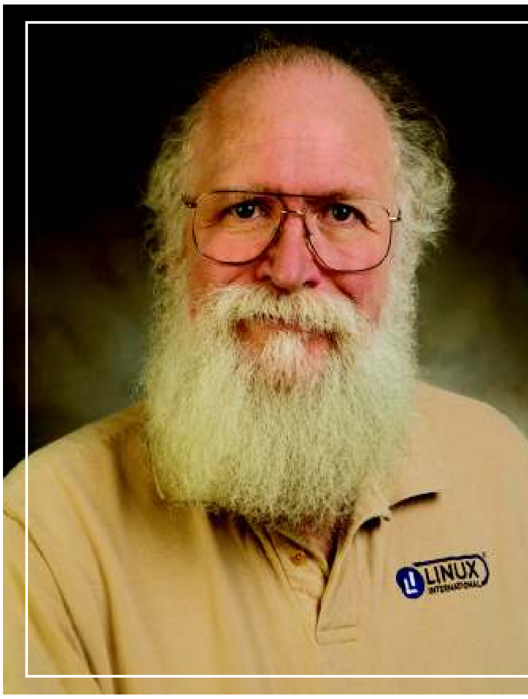
Do not reinvent the wheel

RE-USE

Others have walked a mile (or more) in your shoes.

BY JON "MADDOG" HALL



In 1977 I joined Bell Laboratories in North Andover, Massachusetts and first laid hands on a Unix system. Long before, I had heard of Unix, but even though I had worked on many operating systems, I had never actually seen Unix.

I interviewed at Bell Labs to become a "systems administrator." Beatrice ("B") Fink, who was to be my supervisor, wanted me to be the systems administrator of a CDC Cyber system. As fascinating as that system was, I wanted to be a Unix systems administrator. "The Labs" had developed Unix, and I knew I could find out anything about it by looking at the source code. Besides, Unix ran on PDP-11s and VAX systems from Digital Equipment Corporation (DEC) and had mostly been developed on DEC equipment, and I liked DEC's gear. At the time, I knew nothing about CDC equipment and did not see much of a future in learning about the Cyber.

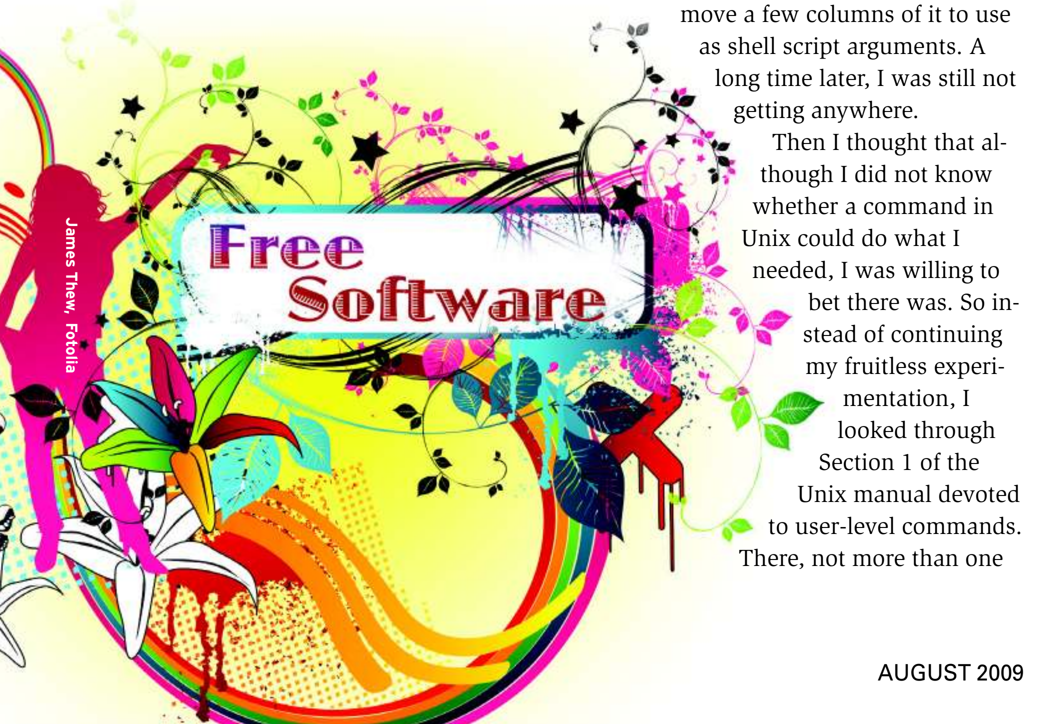
"B" was smart enough to know that it was a losing battle to try and place me with the Cyber, and a couple of days later I was hired to administer the Lab's two Unix systems. Until that time I had worked on operating systems that had, perhaps, two or three levels of directory structure. Programs were typically put in "catalogs," and the "command interpreters" – if they had them – were nowhere near the sophistication of Unix.

I found myself trying to understand and memorize where to put executable files (*/bin*, */usr/bin*, */home/maddog/bin*?), libraries, and other files. I tried to remember the (seemingly) unending number of tiny programs that did so many things.

Fortunately, I had a couple of mentors: Bob Wessling and Tom Merrick, engineers who had been "drafted" into being systems administrators for the Unix systems. They patiently took me along the path of Unix until I began to get my feet under me. For this reason, even many years later, I have sympathy and patience for "newbies" who show desire and effort to learn.

One night I was by myself with the system. By this time I had learned enough shell scripting to be dangerous, and I was trying to take a file and remove a few columns of it to use as shell script arguments. A long time later, I was still not getting anywhere.

Then I thought that although I did not know whether a command in Unix could do what I needed, I was willing to bet there was. So instead of continuing my fruitless experimentation, I looked through Section 1 of the Unix manual devoted to user-level commands. There, not more than one tenth of the way through, was the command *cut(1)*, which did exactly what I needed to do and allowed me to finish my shell script in just a few moments.



James Thew, Fotolia

After that episode, I purposely used a day each quarter to read over the first few sections of the manual, reminding myself of the commands, libraries, and system calls so I would know what functionality was available to me.

Fast forward 32 years to a conversation with a friend. We were talking about a graphics library and the toolkit that came with that environment. A friend of his who did heavy programming with the toolkit would often bemoan not having the needed functionality, only to see that it was planned for the next release or even already there – he just did not know about it. The functionality had appeared because someone else had the same needs he did and so wrote the code; this was the same as I had experienced in 1977.

On SourceForge you have more than 230,000 software projects with more than 2,000,000 registered users. As Free Software continues to grow, these people add functionality, free for the use and reuse of others.

The next time you are stuck with a problem, perhaps the first thing you should do is take the time to see whether the solution already exists: You might find a secret colleague who has traveled that path before you. ■