

ROBERT G. FERRELL

## /dev/random



Robert G. Ferrell is an information security geek biding his time until that genius grant finally comes through.

[rgferrell@gmail.com](mailto:rgferrell@gmail.com)

**I SUPPOSE I'LL ALWAYS REGARD LARRY** Wall as the enlightened soul who finally coaxed my primal programming urge to the fore. Oh, I'd played around with Cobol, Snobol, C, Pascal, Natural, and a few other languages in an idle fashion, the way one might toy with an Elvis-shaped bowtie pasta noodle during a boring testimonial dinner, but I made no commitment to any of them. We both knew it was merely a case of ships passing in the night and nothing lasting would come of it.

I can no longer recall exactly when or where I first encountered Perl. I was using it extensively for CGI scripting by early 1996, so it must have been at least a couple of years prior to that. I was a corporate Webmaster back in the Paleozoic Era of the Web (1994), just after the release of Mosaic. In those days content was still mostly text, Gopher/WAIS was the only real way to search the Web, and the CGI standard hadn't yet been developed. Men were real men, women were real women, and both of us had to trudge for miles uphill both ways through blinding snowstorms to get to the coffee machine every morning. Oh, and our TV remotes only had five or six buttons.

As far as I can remember, the siren song that captivated me where Perl was concerned was that being interpreted rather than compiled meant instant feedback, especially when using the debug command-line option. Prior to that I'd been accustomed to the routine of header files, libraries, and hoping against hope that I'd made the proper ablu-tions to the compiler gods in order to experience the fruits of my coding labors. Now when I made a careless mistake (notice I said "when," not "if") I found out about it immediately and with a reasonable hope of tracing the problem to its source before my next birthday.

Add to that the fact that I actually understood the syntax of Perl on a level not experienced with any previous language and you can probably see why Larry's brainchild was almost on the level of spiritual revelation for me. My mental syntax and Perl's are similar to begin with (snickering allowed only in designated snickering areas, owing to the potential negative health effects of secondhand mirth), so I didn't have to do a lot of translation of thought patterns into formal semantics. This certainly sped things up for me and made troubleshooting programming errors less painful. One curious side ef-

fect was that I began to dream in Perl. This is symbolism that would have driven Sigmund Freud quite sane.

My Perl code is not what you would term “elegant,” unless you employ that same adjective to describe frozen microwave cuisine. It works—most of the time—and that’s about all I feel comfortable asking of myself. I have bought a significantly large fraction of the books ever written about Perl, such as *The Perl Cookbook*, *Object Oriented Perl*, *Perl for Toasters*, and so on, but I never seem to get much out of them. Oh, I might get on a kick and turn out more professional-looking code with fancy-schmancy stuff like comments for a while, but eventually I slide back into my established habits. You can’t teach an old primate new hacks.

Like this CPAN thing. I never really got the hang of using modules. In the time it took me to figure out exactly what each one did and the syntax for using it in my own scripts, I could usually thrash out an equivalent, if less sophisticated, method myself. I guess part of my reluctance to jump on the module bandwagon was due to the fact that the keyboard I had on the AVi-*iON* box where I did most of my coding in the early days of CPAN had a very sticky shift key. That meant that, while the semicolon worked fine, I had a heck of a time getting a colon out of it. To call most module methods you had to type two of those in a row and that was just more trouble than it was usually worth to me. By the time I got around to obtaining a new keyboard (which wasn’t until we went from DG/UX to Solaris 2.5.1), the colon aversion was deeply ingrained.

I realize, of course, that my computer (via its keyboard) was in effect programming *me*. I, for one, welcome our new dopant-laden silicon overlords and wish them better luck in maintaining control over their servants than I’ve ever had. Turn about is, after all, fair play.

Now that I’m past fifty the colon has taken on another, more insidious aspect. Somewhere in middle age a man’s organs just seem to revolt *en masse*, leaving him with a paunch, a mortgage, all the vigor of a store-window mannequin, and kids whose idea of a summer job is taking out the household trash once a week at fifty bucks a pop.

Perl in many ways was the centerpiece of my golden years as a system administrator. I was never a hotshot module jockey, as I’ve hopefully made clear, but I could whip out a one- or two-hundred-line Perl script to solve some pressing sysadmin issue in nothing flat. Last time I looked (which was admittedly about ten years ago), a dozen or so of my scripts were still floating around at Sun’s BigAdmin site. I sent them those scripts, and they shipped me a Sun ballpoint pen. Seemed like a reasonable exchange. At least the pen is still useful.