

# Do You Have a Favorite Shell Trick?

W. Adam Koszek

*Koszek ORG*

wojciech@koszek.com

2022-10-05T17:47:11Z

The very first scripting programming language I learned was Perl. It was circa 2000; Python wasn't that popular back then, and my choice leaned towards Perl since I could get a decent books about it in Polish. I remember squeezing \$20 and getting something that seemed like the thickest book ever.

Several years later I came back to Perl because Xilinx was using it too, and since then I've had the chance to come back and polish some of my Perl chops. We had a very nice build system written in Perl which let you check out files selectively. You would check out a file to a clean directory, hit "build", and it'd build a project for you from vanilla sources. Then you'd check out a file you wanted to modify and keep hacking on it. Next you'd hit "build" again, but this time the system would use the modified file instead of the plain file from the repository. All the junk files from C and Verilog compiler would be put in the obj/ directory for you, so you never had to see them. I liked it, since my workspace was really neat and tidy (two to three source code files plus some log files).

Anyway: the build system was pretty robust. It used a pretty interesting shell trick which lets you make one source file runnable in two or more language interpreters. This came in response to OSes installing Perl in non-standard places and thus making typical `#!` pragma be problematic.

The trick lets your script start as a shell script, and once a Perl interpreter is found, it will "switch" to the interpreter for further execution.

```
#!/bin/sh
#! -*-perl-*-
eval 'exec perl -x -wS $0 ${1+"$@"}'
if 0;
```

If you like it or you have similar issues which this technique could solve, please check out the Polyglot programming page for more information.

**What is your favorite shell trick?**