What is wrong with programming?

M. Simon, Technical Contributor

Poul-Henning Kamp, "one of the primary developers of the FreeBSD operating system", has written a long screed about the current state of programming [1]. He discusses Eric Raymond's *The Cathedral and the Bazaar* (Eric chimes in with a comment) and has this critique of the Bazaar:

> Getting hooked on computers is easy—almost anybody can make a program work, just as almost anybody can nail two pieces of wood together in a few tries. The trouble is that the market for two pieces of wood nailed together—inexpertly—is fairly small outside of the "proud grandfather" segment, and getting from there to a decent set of chairs or fitted cupboards takes talent, practice, and education.

The comments on the piece are voluminous and interesting. Delving into things like make files and autoconf. And those things are problems. But they are not in my opinion the big problem.

I believe the "C" language is the core of our current mess. But people are so used to it that they don't even see it. I'm currently working with a guy who managed 5,000 programmers at AT&T and who is a crack programmer himself and he agrees with me. If all you have is a 75 tool pocket knife and what you really need is a hammer....

As an old timer, I must say that a lot of it is because most programmers these days do not understand the underlying iron. Nor do they understand, nor can they anticipate (other than in a very general way) what the compiler will be doing to their code. "C" is not really conducive to building short well tested fragments as all design books these days tell you is the right way to build code. And argument passing in "C" is clumsy.

The former AT&T guy and I agree that there is probably no way to fix the vast mass of programming out there. Or change the way things are generally done. We are stuck with things as they are. But we both feel that it might be possible to do

something for small projects. With the hope that those ideas eventually percolate "up" into the big stuff.

To get unity of design it is generally agreed that "Cathedral" projects should have one or at most two architects. It is how you get beauty and a clean design. "Designed by a committee" is an epithet. The original "C" design team had two architects. In that case one too many. "C" is not a cathedral. It is a bizarre. You have to build a "prototype" nearly identical to your code so the compiler will understand what to do. Really?

I posted some of these ideas to the thread and a commenter agreed with me:

> I totally agree with above comment. The problem is in the use of "C".

Will we find an answer to all this? Not until the architects start looking for new tools.

Engineering is the art of making what you want from what you can get at a profit.

Source URL (retrieved on *09/18/2014 - 1:27pm*):
http://www.ecnmag.com/blogs/2013/01/what-wrong-programming?qt-most_popular=0

Links:
[1] http://queue.acm.org/detail.cfm?id=2349257