Gibson Research Corporation

Steve Gibson discussing software bugs with Leo Laporte, Security Now episode 302, May 26, 2011.

ref: http://www.grc.com/sn/sn-302.txt

Or listen to the **mp3** of this question.

This is a transcription of a listener question (edited for readability):

LEO: Question #1 comes to us from a listener and programmer. In fact, I know his name, Jim Hyslop. He rants against Steve's claim: Steve, in Episode 256, Q&A #115, your very passionate claim that it is possible to have bug-free software is a major slap in the face to those of us who build software for a living. Well, I'm going to let you defend yourself, Steve. But I think that Steve has been very clear about this, and it's not quite as you say.

Sir, you have stained the honor of all professional programmers, and I will have satisfaction. As the injured party, I hereby challenge you to a thumb wrestling duel to be held at a place and time that is mutually convenient. All joking aside, though, you exclaimed, "Come on, it's math!" No, it's not. Good programming is communication - communicating your intentions to the compiler, and more importantly to other programmers in such a way that there can be no confusion. That's why, as you pointed out in your very much justified rant against JavaScript, every browser interpreted a particular code snippet in a different way. Communication is a human activity, and no matter how carefully you choose your words, someone will interpret what you say in a different manner. I can't count the number of times I've had QA testers file bug reports, only to tell them, "No, that's how it's supposed to work. Look at this section in the design document."

And of course people make mistakes, too. Add in the complexities of multithreaded and event-driven programs, you're now talking about programs whose complexity is several orders of magnitude greater than most programs that could be written in assembler. At that point, proving that a program is bug free is almost an impossible task. You said, "By definition it's possible for us to have an absolutely bug-free environment and not a bug in any apps," and I want to underscore this, "but it'll never happen." You said that. Well, it is also possible to win the lottery five times in a row, but that'll never happen.

Just to be clear, I'm not saying because we can't write bug-free code we should just shrug our shoulders and say "Oh, well," or that we should expect such basic mistakes as not sanitizing inputs, allowing buffer overruns, and so on. Good software developers should, nay, must always do their best to write code that is as bug free as they know how to make it. And yes, Steve, I'm afraid that means there will always be bugs, and people will always make mistakes like shutting off a firewall. Well, in your defense, you've always said that. It's exactly what you've always said.

STEVE: [Laughing] First of all, I also build software for a living.

LEO: You know a little bit about this.

STEVE: Yeah. And he says that "add in the complexities of multithreaded and event-driven programs, and you are now talking about programs whose complexity is several orders of magnitude greater than most programs that could be written in assembler." Well...

LEO: Not so.

STEVE: I write all of my programs in assembler. And they're all multithreaded, and they're all event driven. So I think what he's saying there is that, if you were in assembler, then you're at the bare metal, so you're not depending upon, for example, the whims of the compiler or the JavaScript interpreter and communication. I mean, I certainly agree with him that communication is one of the problems. One of the things I like, for example, about the way the Internet's RFCs, the Requests for Comment, have been structured, is they make a very good point, and they always use all capitals, they say SHALL do this, MUST do this, MAY do this, SHALL NOT, MAY NOT, MUST NOT, you know, they are extremely careful when they're writing these specifications about what behavior they want.

So certainly communication is part of it. Yet I will, at the same time, I will stand by my statement, which is that there isn't anything analog, from the beginning, of the way our computers work. And so, while, yes, it's not going to be the case that massively complex systems written by, when you aggregate all the people involved, tens of thousands of people speaking different languages, thinking different things, meaning different things, and then having it all come together, that it's going to work. In fact, when you phrase it that way, it's amazing these things even boot. *Still, from a theoretical standpoint, and this is really what I was saying in Episode 256, is it can be perfect. It absolutely can be perfect. Will it be? No. Can it be? Absolutely. There's nothing preventing our operating systems and our programs from being perfect. Yeah, they're not going to be. But...*

LEO: Yeah, I mean, in your defense, you've always said that we will never get rid of bugs.

STEVE: Right.

LEO: So there's a difference between making an assertion that theoretically it is possible, as with anything like this, to make it perfect. But practically it's impossible.

STEVE: Correct.

LEO: Would that be a fair way to describe what you believe?

STEVE: Yes. Practically, it's, well, practically it's never going to happen. It's not impossible, it's never going to happen. I love Donald Knuth's book on, is it LeX?

LEO: No, TeX. TeX, it's called, T-e-X.

STEVE: Yeah, TeX.

LEO: He pronounces it "Tech," just to be additionally obscure.

STEVE: And in the preface he says, "I believe on" - and he quotes a date - "the last bug in this program was found." And I think he says, "And the person who found it got $2.56. And I will double the reward for every successive bug that's found, except I don't think there are anymore."

LEO: Wow. That's bold.

STEVE: And I love that. No, I mean, again, I mean, this is Donald. I mean, this is the guy - he's an artist of software.

LEO: That's the name of his book.

STEVE: And TeX is massive. I mean, it is a serious piece of code. And it may very well be perfect.

LEO: So no one's found another bug?

STEVE: No. There probably aren't any.

LEO: Wow.

STEVE: I mean, he probably did find them all. Because he wrote it in a language that he knew, well, actually in a language that he invented, and wrote it very carefully, and, I mean, that's just the way he is. Now, is that a commercial practicality? No. I mean, he would have been fired by any employer. But he's the person we all bow to as the master of the art and science of computer programming. Not the economics, not the practical reality. But boy, he knows what he's doing. And he wrote a perfect program.