The Embedded Muse 17

Editor: Jack Ganssle (jack@ganssle.com) October 17, 1997

Editor's Notes

Today's issue of the Muse is a bit off-format. Somehow, from somewhere, propagated over the ether of the net in an unending stream of anonymity, the following interview appeared in my IN-box.

Though the article is a joke, it reinforces my concerns with using any new language for embedded work. Till a standard exists, and till a cadre of well trained programmers are at hand, it's risky to bet on any new lingo.

C++ brings many potential benefits to the embedded world, as well as its own set of baggage. A recent article in Embedded Systems Programming magazine addressed some of these issues and discussed an alternative - EC++, a version designed specifically for embedded systems where a lighter footprint is essential. EC++ preserves most of the neat stuff about C++ while stripping out high-overhead things like multiple inheritance.

So, since the issue of languages seems to bring out the flame wars… and since it's dangerous to take ourselves too seriously, here it is:

An Interview with Bjarne Stroustrup

Interviewer: Well, it's been a few years since you changed the world of software design, how does it feel, looking back?

Stroustrup: Actually, I was thinking about those days, just before you arrived. Do you remember? Everyone was writing 'C' and, the trouble was, they were pretty damn good at it. Universities got pretty good at teaching it, too. They were turning out competent - I stress the word 'competent' - graduates at a phenomenal rate. That's what caused the problem.

Interviewer: Problem?

Stroustrup: Yes, problem. Remember when everyone wrote Cobol?

Interviewer: Of course, I did too

Stroustrup: Well, in the beginning, these guys were like demi-gods. Their salaries were high, and they were treated like royalty.

Interviewer: Those were the days, eh?

Stroustrup: Right. So what happened? IBM got sick of it, and invested millions in training programmers, till they were a dime a dozen.

Interviewer: That's why I got out. Salaries dropped within a year, to the point where being a journalist actually paid better.

Stroustrup: Exactly. Well, the same happened with 'C' programmers.

Interviewer: I see, but what's the point?

Stroustrup: Well, one day, when I was sitting in my office, I thought of this little scheme, which would redress the balance a little. I thought 'I wonder what would happen, if there were a language so complicated, so difficult to learn, that nobody would ever be able to swamp the market with programmers?' Actually, I got some of the ideas from X10, you know, X windows. That was such a bitch of a graphics system, that it only just ran on those Sun 3/60 things. They had all the ingredients for what I wanted. A really ridiculously complex syntax, obscure functions, and pseudo-OO structure. Even now, nobody writes raw X-windows code. Motif is the only way to go if you want to retain your sanity.

Interviewer: You're kidding...?

Stroustrup: Not a bit of it. In fact, there was another problem.. Unix was written in 'C', which meant that any 'C' programmer could very easily become a systems programmer. Remember what a mainframe systems programmer used to earn?

Interviewer: You bet I do, that's what I used to do.

Stroustrup: OK, so this new language had to divorce itself from Unix, by hiding all the system calls that bound the two together so nicely. This would enable guys who only knew about DOS to earn a decent living too.

Interviewer: I don't believe you said that...

Stroustrup: Well, it's been long enough, now, and I believe most people have figured out for themselves that C++ is a waste of time but, I must say, it's taken them a lot longer than I thought it would.

Interviewer: So how exactly did you do it?

Stroustrup: It was only supposed to be a joke, I never thought people would take the book seriously. Anyone with half a brain can see that object-oriented programming is counter-intuitive, illogical and inefficient.

Interviewer: What?

Stroustrup: And as for 're-useable code' - when did you ever hear of a company re-using its code?

Interviewer: Well, never, actually, but...

Stroustrup: There you are then. Mind you, a few tried, in the early days. There was this Oregon company - Mentor Graphics, I think they were called - really caught a cold trying to rewrite everything in C++ in about '90 or '91. I felt sorry for them really, but I thought people would learn from their mistakes.

Interviewer: Obviously, they didn't?

Stroustrup: Not in the slightest. Trouble is, most companies hush-up all their major blunders, and explaining a $30 million loss to the shareholders would have been difficult. Give them their due, though, they made it work in the end.

Interviewer: They did? Well, there you are then, it proves O-O works.

Stroustrup: Well, almost. The executable was so huge, it took five minutes to load, on an HP workstation, with 128MB of RAM. Then it ran like treacle. Actually, I thought this would be a major stumbling-block, and I'd get found out within a week, but nobody cared. Sun and HP were only too glad to sell enormously powerful boxes, with huge resources just to run trivial programs. You know, when we had our first C++ compiler, at AT&T, I compiled 'Hello World', and couldn't believe the size of the executable. 2.1MB

Interviewer: What? Well, compilers have come a long way, since then.

Stroustrup: They have? Try it on the latest version of g++ - you won't get much change out of half a megabyte. Also, there are several quite recent examples for you, from all over the world. British Telecom had a major disaster on their hands but, luckily, managed to scrap the whole thing and start again. They were luckier than Australian Telecom. Now I hear that Siemens is building a dinosaur, and getting more and more worried as the size of the hardware gets bigger, to accommodate the executables. Isn't multiple inheritance a joy?

Interviewer: Yes, but C++ is basically a sound language.

Stroustrup: You really believe that, don't you? Have you ever sat down and worked on a C++ project? Here's what happens: First, I've put in enough pitfalls to make sure that only the most trivial projects will work first time. Take operator overloading. At the end of the project, almost every module has it, usually, because guys feel they really should do it, as it was in their training course. The same operator then means something totally different in every module. Try pulling that lot together, when you have a hundred or so modules. And as for data hiding. God, I sometimes can't help laughing when I hear about the problems companies have making their modules talk to each other. I think the word 'synergistic' was specially invented to twist the knife in a project manager's ribs.

Interviewer: I have to say, I'm beginning to be quite appalled at all this. You say you did it to raise programmers' salaries? That's obscene.

Stroustrup: Not really. Everyone has a choice. I didn't expect the thing to get so much out of hand. Anyway, I basically succeeded. C++ is dying off now, but programmers still get high salaries - especially those poor devils who have to maintain all this crap. You do realize, it's impossible to maintain a large C++ software module if you didn't actually write it?

Interviewer: How come?

Stroustrup: You are out of touch, aren't you? Remember the typedef?

Interviewer: Yes, of course.

Stroustrup: Remember how long it took to grope through the header files only to find that 'RoofRaised' was a double precision number? Well, imagine how long it takes to find all the implicit typedefs in all the Classes in a major project.

Interviewer: So how do you reckon you've succeeded?

Stroustrup: Remember the length of the average-sized 'C' project? About 6 months. Not nearly long enough for a guy with a wife and kids to earn enough to have a decent standard of living. Take the same project, design it in C++ and what do you get? I'll tell you. One to two years. Isn't that great? All that job security, just through one mistake of judgment. And another thing. The universities haven't been teaching 'C' for such a long time, there's now a shortage of decent 'C' programmers. Especially those who know anything about Unix systems programming. How many guys would know what to do with 'malloc', when they've used 'new' all these years - and never bothered to check the return code. In fact, most C++ programmers throw away their return codes. Whatever

happened to good ol' '-1'? At least you knew you had an error, without bogging the thing down in all that 'throw' 'catch' 'try' stuff.

Interviewer: But, surely, inheritance does save a lot of time?

Stroustrup: Does it? Have you ever noticed the difference between a 'C' project plan, and a C++ project plan? The planning stage for a C++ project is three times as long. Precisely to make sure that everything which should be inherited is, and what shouldn't isn't. Then, they still get it wrong. Whoever heard of memory leaks in a 'C' program? Now finding them is a major industry. Most companies give up, and send the product out, knowing it leaks like a sieve, simply to avoid the expense of tracking them all down.

Interviewer: There are tools...

Stroustrup: Most of which were written in C++.

Interviewer: If we publish this, you'll probably get lynched, you do realize that?

Stroustrup: I doubt it. As I said, C++ is way past its peak now, and no company in its right mind would start a C++ project without a pilot trial. That should convince them that it's the road to disaster. If not, they deserve all they get. You know, I tried to convince Dennis Ritchie to rewrite Unix in C++.

Interviewer: Oh my God. What did he say?

Stroustrup: Well, luckily, he has a good sense of humor. I think both he and Brian figured out what I was doing, in the early days, but never let on. He said he'd help me write a C++ version of DOS, if I was interested.

Interviewer: Were you?

Stroustrup: Actually, I did write DOS in C++, I'll give you a demo when we're through. I have it running on a Sparc 20 in the computer room. Goes like a rocket on 4 CPU's, and only takes up 70 megs of disk.

Interviewer: What's it like on a PC?

Stroustrup: Now you're kidding. Haven't you ever seen Windows '95? I think of that as my biggest success. Nearly blew the game before I was ready, though.

Interviewer: You know, that idea of a Unix++ has really got me thinking. Somewhere out there, there's a guy going to try it.

Stroustrup: Not after they read this interview.

Interviewer: I'm sorry, but I don't see us being able to publish any of this.

Stroustrup: But it's the story of the century. I only want to be remembered by my fellow programmers, for what I've done for them. You know how much a C++ guy can get these days?

Interviewer: Last I heard, a really top guy is worth $70 - $80 an hour.

Stroustrup: See? And I bet he earns it. Keeping track of all the gotchas I put into C++ is no easy job. And, as I said before, every C++ programmer feels bound by some mystic promise to use every damn element of the language on every project. Actually, that really annoys me sometimes, even though it serves my original purpose. I almost like the language after all this time...

Interviewer: You mean you didn't before?

Stroustrup: Hated it. It even looks clumsy, don't you agree? But when the book royalties started to come in... well, you get the picture...

Interviewer: Just a minute. What about references? You must admit, you improved on 'C' pointers...

Stroustrup: Hmm. I've always wondered about that. Originally, I thought I had. Then, one day I was discussing this with a guy who'd written C++ from the beginning. He said he could never remember whether his variables were referenced or dereferenced, so he always used pointers. He said the little asterisk always reminded him...

Interviewer: Well, at this point, I usually say 'thank you very much' but it hardly seems adequate...

Stroustrup: Promise me you'll publish this. My conscience is getting the better of me these days...

Interviewer: I'll let you know, but I think I know what my editor will say...

Stroustrup: Who'd believe it anyway? Although, can you send me a copy of that tape?

Interviewer: I can do that...

About The Embedded Muse

The Embedded Muse is an occasional newsletter sent via email by Jack Ganssle. Send complaints, comments, and contributions to him at jack@ganssle.com.

To subscribe, send a message to majordomo@ganssle.com, with the words "subscribe embedded *your-email-address*" in the body. To unsubscribe, change the message to "unsubscribe embedded *your-email-address*".

The Embedded Muse is supported by The Ganssle Group, whose mission is to help embedded folks get better products to market faster. We offer seminars at your site offering hard-hitting ideas - and action - you can take now to ***improve firmware quality and decrease development time***. Contact us at info@ganssle.com for more information.