

Re: A petition to J3 apropos FORTRAN's future

Source: <http://coding.derkeiler.com/Archive/Fortran/comp.lang.fortran/2004-03/1182.html>

From: James Giles (jamesgiles_at_worldnet.att.net)

Date: 03/22/04

Date: Mon, 22 Mar 2004 20:34:33 GMT

analyst41 wrote:

> "James Giles" <jamesgiles@worldnet.att.net> wrote in message
> news:<[t2o7c.12802\\$PY1.289534@bgtnc05-news.ops.worldnet.att.net](mailto:t2o7c.12802$PY1.289534@bgtnc05-news.ops.worldnet.att.net)>...
>> "Grown up"? I guess that's why most C code looks like it was
>> written by a talentless eight year old?
>
> It sure does – but since Capitalism drives everything in the world,
> let us give C (to me C means C/C++/JAVA) its due – C being the
> foundation of Microsoft ORACLE etc. – not to speak of innumerable C
> applications (including quantitative ones) running in tax-paying
> corporations does mean something.
>
> C won the ultimate contest – it gets used. If we refuse to see that C
> did something right – then thats truly childish.

If you misidentify what C supposedly did right, you're truly counterproductive. C was in the right place at the right time. That's pretty much all it did right.

> That C has (probably) irreparably decimated Fortran's user base is a
> fact. If my reasons why are incorrect – we should find the true
> reasons and discover is there any way the user base can be brought
> back. A REALLY tired argument in this regard would be "irrationality
> of decision makers in software projects".

But, the fact that an argument is REALLY tired doesn't make it false. And yes, the decisions were taken irrationally. The most popular reason given was that C was so common (this reason was used clear back in the late 70's – when almost no one was using C, but that didn't deter the promoters of C from using the claim). Other promotional C dogmas were as easily dismissed. There was never a valid *technical* reason to use C.

Ironically, if you had a rational plan for improving Fortran I might even support the effort. I too think the committee has gone in some wrong directions since the 80's.

But, trashing the lot and adopting even *worse* features instead is really not a viable choice. Discarding features very selectively and discussing genuinely well-designed replacements might be productive. It would have to be a separate language: no standardized language can undo design decisions in a timely fashion even if *everyone* agreed they were mistakes.

So, what are your *specific* recommendations about what features to delete? And, more important, what are your *specific* recommendations for what to do instead? Just saying we need more "dirty features" doesn't address the question. What "dirty features"? Exactly how would they work? How would they fit in with the parts of Fortran you plan to keep?

--

J. Giles