Re: Larkin, Power BASIC cannot be THAT good:

Source: http://sci.tech−archive.net/Archive/sci.electronics.design/2009−06/msg03317.html

- *From*: Nobody <nobody@xxxxxxxxxxx>

- *Date*: Sat, 20 Jun 2009 23:15:39 +0100

On Sat, 20 Jun 2009 08:43:46 −0700, John Larkin wrote:

> For the project which is in front of me right now, out of 53 C source files (not including headers), 2 are Unix−specific, 2 are Windows−specific, the other 49 are portable. There are precisely 6 #if[def] directives in the entire code (mostly for platform−specific headers), and no platform−specific macros.

> 53?! All the programs I've done in the last few years have each had one source file. The only exception is embedded things that have one uP code source file and one to four FPGA configuration files, which all get built into one rom image file.

Another package which I work on has ~2500 C files, 60 C++ files, 160 Python files and 550 Makefiles, resulting in 350 executables and 50 libraries. And that's not all that large; the numbers are inflated by virtue of it being composed of many small modules.

Okay, so this isn't "embedded" software, or even "system" software. But the Linux kernel is both of those, and is far bigger (10,000 C files, 1000 assembler files, although not all files will be used on any particular architecture).

At that level, things like structure, abstraction, and the development process matter a lot. You can't rely upon programmers understanding the whole thing, or no−one ever making a mistake (particularly when so much of it is for hardware which is inadequately or inaccurately documented). And testing doesn't help much when much of the bug potential is related to concurrency.

> what
> difference does it make what language it's in before you rewrite it?
> From C to C, or C++, or Java, or to many of the other braces type

> > languages, you might only need to refactor some keywords
> > to make it
> > fit.
>
> Or you might (and probably will) need to completely rewrite it. C doesn't have classes, Java doesn't have pointers, or half of C++'s OO features, or many of the libraries available to C/C++ code. C and C++ don't have garbage collection.

> So the only languages that allow 40−year old programs to be simply tweaked to compile and run on modern OSs are Fortran, Basic, and Cobol.

Not at all. You would have just as many problems converting Fortran, Basic or Cobol to each other or to C, C++ or Java. IOW, the claim of "only need to refactor some keywords" is bogus.

.