

---

Subject: Re: "(national) C compiler"

Posted by [cbpporter](#) on Wed, 01 Apr 2009 14:43:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

kbyte wrote on Wed, 01 April 2009 17:07I just plain to attract people to program their programs...  
helder people, young people, people that do not understand english...  
May be I am following a wrong strategy...

Thank you very much

Kim

Well if this is your objective you may be following the wrong strategy.

First of all, there is no good reason for non professionals to learn C. I can be even damaging. While the core of the language is as fresh as ever and will remain like that forever (I mean writing algorithms, reasoning and principles, and even syntax), other features are very dated and counter productive. C is not something to which you want to attract people. C is what you do when you need to work on embedded systems, need extreme performance code (but here also C++ does the job as fast), need very fast compile speed by careful structuring of you code, need dynamic libraries, need to create a new compiler by having it generate C and thus opening up all existing platforms and so on. If you want to attract people to programming try some scripting language, ObjectPascal or Java. ObjectPascal is based on Pascal, which has didactic roots and while being a competent tool it makes it very hard to shoot yourself in the proverbial foot. But don't try Turbo Pascal, because people generally hate it .

As for using native keywords to make learning easier and more attractive, I don't think it's going to count for C. C is not a verbose language. You use about 5 keywords very often in an imperative way: if, else, while, return and class. And even if you add the rest, their number is quite low. These could benefit from native language form, but still people will grasp quite easily "if" even if they don't understand the word. As for the rarer ones like volatile and register, I doubt that by renaming them you will make the concept easier to grasp. On the other hand, "char \*(\*(\*\*foo [[8]])())[];" will look just as but no matter what you put in the place of char. And here unfortunately preprocessor wont help. And you can use typedef to replace all the keywords that define types.

---