
Subject: Re: "(national) C compiler"

Posted by [mr_ped](#) on Wed, 01 Apr 2009 13:56:13 GMT

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

depends what you hit.

{ and } are IMHO ideal for macro (although maybe I overlook something, but it looks very safe to me right now) except when used inside macros themselves.

#include is probably impossible to change (outside of compiler), because it's directive of preprocessor, thus it is handled before the translation in preprocessor begins. (basically any preprocessor directive will be difficult to change)

In this case it would better to create your own native language -> C compiler.

I'm still missing the point of this effort.

Such verbose language exists, it's Pascal, and the only reason why I did code simple application faster in Pascal then in C was that Borland Pascal did compile executable usually under 3 seconds while Borland C++ took often over minute. Otherwise having "begin end" make productivity suffer IMHO. You have to write it longer, and in the end I find the {} even easier to read and to match (in well intended code), so I think it's better to spend couple of hours to train somebody to recognize { } without second thought, then having "begin" "end" in programming language.

Then there are C keywords. There's only handful of them, and knowing the C original allows you to use C compilers (and they are many and almost everywhere). Any knowledge of native translation of them will leave you in very impractical position of knowing something what is not used anywhere except your special case.

And sources of application usually consist lot more of calls to library functions, then C keywords. Unless you plan to create wrapper for every common library to have their native language names, most of your application will look far from native language, only the keyword part of them.

I'm not saying it's not good to see your algorithm in native language first, but in my 15+ years experience I find paper+pencil much more stronger then any programming language. After I sort this one out, I simply translate it into C/C++ or other programming language in my head, but usually (for me) it's far easier to work with paper then computer during the design of algorithm phase.

But I think it's wasting someone's time, to make him learn something as impractical as native language version of C.
