Subject: Re: "(national) C compiler"
Posted by cbpporter on Wed, 01 Apr 2009 07:28:13 GMT
View Forum Message <> Reply to Message

kbyte wrote on Wed, 01 April 2009 00:03
So, what do you to advice me?

Well, I wouldn't invest effort into this except with macros. You could parse the compiler errors and with basic regex get it to talk about your keywords.

But if you want a "true" solution, I still suggest taking a C compiler and just changing what the keywords are. If you're lucky, you wont have to edit some lexer tool generated tables. LCC and clang are good candidates. If you're feeling brave, you could try to tackle GCC. GCC may be complex and not very coder friendly (it's sources), but changing the lexer should be accessible.

Quote:Actually, C is pretty simple. I believe I could do basic C compiler in a week (badly optimizing, compiling to e.g. x86 assembly).

On the other side, C++ is complex - I would say year is not enough to implement it (just to showcase relative difficulity).
Sure, one week is doable. As I said, 3000-5000 lines as a rough estimate and with a back end to handle platform specific code generation you could even have something relatively cross platform. Clang is significantly larger than this because it is modular and tries do do a lot of fancy stuff, like keeping all precompiler information so it can generate meaningful compiler errors based on source code before precompilation. Also tries to emulate GNU extensions.

But C is IMO very complex, not from a pure language point of view, but from an implementation point of view. You have to get a lot of features together and working before your compiler can do anything. Just to write a compiler that can genuinely handle the requirements of hello world, you need to implement the preprocessor, pointers, strings with at least one escape character and functions with variable argument list. A lot of small features, but these do ramp of the complexity in the initial stages. With other languages, you could get simple cases working with considerably less effort. Also, operators in C are not that easy to implement with optimizations because C accepts a wide range of combinations and expressions which are hardly readable by human being sometimes make their way into production code.