

Re: Announcing the RosAsm Library Project

Source: <http://coding.derkeiler.com/Archive/Assembler/alt.lang.asm/2007-10/msg00293.html>

- *From:* hutch-- <hutch@xxxxxxxxx>
 - *Date:* Thu, 04 Oct 2007 16:50:15 -0700
-

This is in fact a very good point. For any code base to be useful it must have a basic set of components that can be used for the simple hacky stuff and this is of course what static libraries provide among many other things. Another factor that is missed by people who are not familiar with the power and flexibility of libraries is that with a common object module format "COFF" in the instance of Windows code is fully portable across different assemblers in the form of object modules.

You can write an object module in GAS and use it in a FASM application, a POASM module in MASM, a HLA module in GAS and so on which makes the object module capacity a very powerful and flexible one.

It is in fact easy enough to have a snippet collection that can be cut and pasted into a program being developed but in that form its limited to a single depth call tree where libraries and linkers can calculate and depth of dependency and get it right without any duplication of code whatsoever.

For languages that don't have a static library / linker capacity they need to have a very good equate and conditional assembly capacity and then provide a matching capacity by the use of conditional includes.

It is not a big deal technically to have to do either, if you have a reasonable hash table procedure available you just set a non executable flag (usually an equate) in the text code module if it needs another text module as a dependency and when all of the initial calls are processed there will also be a list of dependencies which can be included as well.

This is not as simple to create at the user end but it is a viable mechanism for people like Betov who simply don't know how to create an object module or a linker. It would simply mean that if a RotAsm user called a procedure from this text based library, she would automatically get the dependencies that are required and the program would build without any missing components.