Subject: GCC code size optimizations on ARM - GNU/Linux - uClibC
Posted by chickenk on Wed, 18 Jun 2008 16:39:04 GMT
View Forum Message <> Reply to Message

Hi everyone,

I am currently using theIde as a cross platform IDE for my embedded platform based on an arm926t core, running linux. So far so good, theIde is very useful and easy to configure to use a cross-toolchain.

I made a few test in compiling sample apps based on the Core library as well. Using my arm-linux-uclibc toolchain and a few customizations in the Core library to avoid MMX/SSE detection and other assembly code, I get a 456kB output file for the Core06 example.

In an attempt to reduce the output binary code size as much as possible, I added -ffunction-sections to the compiler options and --gc-sections to the linker options. It used to be a great solution for me on some previous projects. All I got is... 453kB. Not much better. That also means that the source files are well defined, since wrapping sections over object files or over functions is quite the same.

Then I added -fdata-sections to the compiler options, and this time, the size is reduced to 260kB. Great improvement, but makes me wonder:

does anybody here suspect this optimization to make my apps miserably fail later ? I will make tests of course (I'm sharing the device platform so I cannot right now, but tomorrow probably), but I may miss something that someone here already knows about the U++ library that needs this option not to be set.

Anyway, I'll try to give you informed of my results, if anyone interested...

Lionel