
Scientific Computing 1
Handout 3
November 2, 2014

Common GCC Options

Binary code optimization:

-Os	Optimize the code to reduce the size of the binary.
-O1	Turn on basic optimizations. The compiler tries to reduce code size and execution time, without performing any optimizations that take a great deal of compilation time.
-O2	Optimize even more. GCC performs nearly all optimizations that do not involve a space-speed trade-off. As compared to -O1, this option increases both compilation time and the performance.
-O3	Aggressive optimization. It tries to unroll loops constructs and inlines small functions. It can cause unexpected effects in the program. The output is usually larger than using -O2.
-march=native	Automatically determines the code generation options to optimally exploit your local CPU features. Code may not be executable on other machines.

Debugging:

-g	Include the debug symbols in the output. This is necessary for tools like gdb, ddd or valgrind.
-pg	Include the profiling information for the GNU profiler.

Floating Point Arithmetics related:

-ffast-math	Turns off the IEEE754 floating point arithmetics. This option is dangerous.
-ffloat-store	Floating point operations store the results to the memory instead of keeping them in high accuracy CPU registers.
-mfpmath=sse -msse2	Use the SSE2 registers for floating point operations instead of the classical x86/x87 floating point unit. Only available on x86 and x86_64 platforms.

Warnings and C Standards:

-Wall	The compiler displays all warning about malformed code.
-std=XXX	Defines the C standard to use. Normally this is not necessary, e.g.: c89, c99 or c11.

Finding libraries and header files:

-Ipath	Set an additional search path for the include directive. This can be used multiple times.
--------	---

<code>-Lpath</code>	Set an additional search path for the linker.
<code>-lNAME</code>	Link a specified library to the program. The <code>lib</code> prefix is automatically added to the library.

Compilation of own libraries:

<code>-c</code>	Compile the source code to object files without linking it. The default output name is <code>inputname.o</code> .
<code>-fPIC</code>	Generate <i>position independent code</i> . This flag influence the assembler code production to use relative addresses. It is necessary for libraries.

Code Preprocessing and basic shared memory parallelism:

<code>-DNAME=VALUE</code>	Defines a preprocessor variable <code>NAME</code> and sets it to <code>VALUE</code>
<code>-fopenmp</code>	The OpenMP support is enabled.
<code>-pthread</code>	The PThread support is enabled.