Compiler diagnostic flags

Portability and correctness both are important goals when developing code. Non-standard code may not be portable, and its execution may be unpredictable.

Using diagnostic options when you compile your code can help you find potential problems. Since the compiler is going to analyze your code anyway, it pays to take advantage of the diagnostic options to learn as much as you can from the analysis. Please note that some compilers disable the default optimization when you switch on certain debugging flags.

Because of differences in compilers, it also is good practice to compile your code with each compiler that is available on the system, note any diagnostic messages you get, and revise your code accordingly.

The following options can be helpful as you compile code to run in the HPC environment that CISL manages.

Page contents

Cray

The following compiler flags may be helpful for debugging your code:

- **-G0 –** provide complete debugging information with optimizations disabled (i.e. **-O0**, **-O ipa0**, **-O scale0**, **-O vector0**). Breakpoints can be set at different sections of the code for easier debugging.
- **-G01 –** generate debugging report with partial optimization.
- **-G02 –** generate debugging report with full optimization.
- **-g –** generate debugging report (equivalent to **-G0**).
- **-h bounds** - Enables checking of array bounds, pointer and array references at runtime.

Also see Cray C/C++ debug options.

Intel

- **-debug all** – provides complete debugging information.
- **-g** – places symbolic debugging information in the executable program.
- **-check all** – performs all runtime checks (includes bounds checking).
- **-warn all** – enables all warnings.
- **-stand f08** – warns of usage that does not conform to the Fortran 2008 standard.
- **-traceback** – enables stack trace if the program crashes.

Also see Intel C++ diagnostic options.

GNU

- **-ggdb** – places symbolic debugging information in the executable program for use by GDB.
- **-fcheck=all** – performs all runtime checks (includes bounds checking).
- **-Wall** – enables all warnings.
- **-std=f2008** – warns of usage that does not conform to the Fortran 2008 standard.

Also see GCC diagnostic warning options.

NVIDIA HPC SDK

The following compiler flags may be helpful for debugging your code using NVIDIA HPC SDK.

- **-g** – Include symbolic debugging information in the object modules with optimization disabled (**-O0**).
- **-gopt** – Include symbolic debugging information in the object modules without affecting any optimizations.
- **-C or -Mbounds** – Add array bounds checking.
- **-Mchkptr** – Check for unintended de-referencing of NULL pointers.

- **-Minform=inform** - Display all the error messages of any severity (inform, warn, severe and fatal) during compilation phase.

Also see NVIDIA HPC SDK documentation.