

Introduction to Linux Building Software

Peter Ruprecht

peter.ruprecht@colorado.edu

www.rc.colorado.edu

Outline

- Steps from source to executable
- Basic compiling and linking
- Creating and linking libraries
- Optimization and performance libs
- Parallel programming
- Makefiles
- Profiling
- Debugging

From Source to Executable

- Create source code in one or more text files.
- Preferred languages are C/C++, and Fortran 90/95
- Optionally preprocess (cpp, fpp)
- Compile/assemble into object files (gcc, g++, gfortran, icc, icpc, ifort)
- Link object files, possibly along with additional libraries, into an executable
- The last two steps are often combined for simple programs

Additional Steps

- Prepare input files if needed
- Execute the program and check its accuracy and performance
- Use a debugger if necessary to find why program doesn't work as expected
- Improve its performance by recompiling with optimization or profiling flags

Compile/Link Commands

- `f90 my.f`
 - (compile, link; resulting executable named `a.out`)
- `f90 -o my.x my.f`
 - (compile and link `my.f` and name executable `my.x`)
- `f90 -c my.f`
 - (only compile `my.f`, producing object file `my.o`)
- `f90 -o my.x my.f -lnag -L/central/lib`
 - (same as second, but also link in `libnag.a` library, finding it in `/central/lib`)
 - (In C, you may need to specify the location of header or "include" or ".h" files with the `-I` flag)

Compile/Link, cont'd

- `f90 -o my.x a.o b.o`
 - (link a.o, b.o object files, to create executable file my.x)
- `f90 -c -o my.x -O4 a.f b.f`
 - (allows for interprocedural optimization, thus faster execution)
- `f90 -ipo -o my.x a.f b.f`
 - (using compiler's built-in interprocedural optimization; better)
- `f90 -o my.x -g -O0 a.f b.f`
 - (sets up my.x for later debugging)

Libraries

- Like object files; contain reusable functions; convenient
- Often highly optimized for performance
- Archive (.a) and shared (.so); static vs dynamic linking
- -l`starpac` links in `libstarpac.a` or links in references to the functions in `libstarpac.so`
- -L/path/to/libraries tells the linker what directory to search for libs
- Order of linking can be important!
- nm lists the functions defined in a library
- ar creates an archive library