Lab #5: GNU Make

PCSE 2015

Lab assignment

1. Download the Fortran and/or C files for Lab 5. Try using the `make` command in the same directory as the Makefile and sources. What command-line output do you get? What new files are generated?
2. Try running `make clean`. What command-line output do you get? What files are present?
3. Run `make` again. And then run `make` one more time. Did anything change on the command-line output? If so, what? What about any changes to the files?
4. In the Makefile, at the very top of the file, add a comment with your name and the date created. Try running `make clean` and `make`. Does it still work?
5. Just below your comment, add a new `info` target. The target should echo the three intended possible `make target` commands. We have already used `make` and `make clean`. What is the third target we would use to make our executable? Hint: The target name is language dependent in our little example. Make sure to use tabs!
6. Create a variable "CC" or "FC" for your compiler. Instead of using "gcc" or "gfortran", set your new variable to Intel's compilers "icc" or "ifort". Substitute this new variable where apropriate. What does your Makefile look like now? Try re-running `make clean` and `make fooexe` where "fooexe" is the target from the previous question. Do they still work?
7. Do `make clean`. Try moving foomod.F90 to boomod.F90 and/or foo.c to boo.c. Run `make fooexe` again. What error did you get? Explain the error. Be sure to move boomod.F90 and/or boo.c back to foomod.F90 and/or foo.c.
8. Run `make clean`. Move your Makefile to My_Makefile. Run `make fooexe`. What output do you get? Why?
9. On the command-line, type `man make`. Find the OPTION that you can add to your make command to accept a different makefile name. Type `q` to quit the

man page for make. What is the exact command-line you would use to create target fooexe with a makefile called My_Makefile? Move "My_Makefile" back to "Makefile".

10. Run `make clean`. You can change variables within the Makefile from the command-line. The syntax is `make fooexe VARIABLE=VALUE` Try changing your compiler variable so it is gcc and/or gfortran again. Try changing your comiler variable on the command-line so it also has a `-O3` optimization flag. What does that syntax look like? What is your output from running this command?
11. Fortran folks only: add an `OPT` variable inside your makefile. Set it to `-O3`. Appropriately place the variable on the compile lines. What does your Makefile look like now?
12. C folks only: add a `CFLAGS` variable inside your makefile. Set it to the math library `-lm`. Appropriately place the variable on the link line to substitute the `-lm` that already exists. What does your Makefile look like now?
13. Use the % template syntax along with the $⟨ automatic variable to collapse your two compile targets into one. Fortran folks: You'll have to add an additional rule to satisfy foomain.o's dependcy on foomod.o. You can add the rule as `foomain.o : foomod.o`. What does your Makefile look like now?