**Makefile Assignment**

The GitHub repo contains four files: driver.c, file1.c, file2.c, and headers.h. You are to create a makefile that builds the project when you enter the command "make all" at the command line or "make makefileAssignment":

matt@trout:makefileAssignment$ make all

clang -c -o driver.o driver.c

clang -c -o file1.o file1.c

clang -c -o file2.o file2.c

clang -o makefileAssignment driver.o file1.o file2.o

matt@trout:makefileAssignment$

In addition, if a change is made to driver.c (perhaps via the *touch* command), "make all" will cause driver.c to be compiled and then the executable, makefileAssignmet, will again be created via linking:

matt@trout:makefileAssignment$ touch driver.c

matt@trout:makefileAssignment$ make all

clang -c -o driver.o driver.c

clang -o makefileAssignment driver.o file1.o file2.o

matt@trout:makefileAssignment$

Notice that ONLY driver.c is recompiled, not file1.c or file2.c, because they were not changed since their corresponding object files were created.

The same behavior should be true for file1.c and file2.c. For example, suppose both of those files are changed:

matt@trout:makefileAssignment$ touch file1.c file2.c

matt@trout:makefileAssignment$ make all

clang -c -o file1.o file1.c

clang -c -o file2.o file2.c

clang -o makefileAssignment driver.o file1.o file2.o

matt@trout:makefileAssignment$

Again, notice that driver.c was not recompiled, because its corresponding object file, driver.o, is newer than its prerequisite driver.c.

Lastly, if headers.h is changed, all the .c files should be recompiled and the executable recreated, since all the .c files #include headers.h. The modification of your makefile's rule(s) to accomplish this is hinted at in the lecture slides.

matt@trout:makefileAssignment$ touch headers.h

matt@trout:makefileAssignment$ make all

clang -c -o driver.o driver.c

clang -c -o file1.o file1.c

clang -c -o file2.o file2.c

clang -o makefileAssignment driver.o file1.o file2.o

matt@trout:makefileAssignment$

Your makefile should also include a rule with a (phony) target, clean. When this rule triggers via a command line "make clean", it should remove all the .o files and the executable makefileAssignment. Example:

matt@trout:makefileAssignment$ ls

Makefile file1.c headers.h

build file1.o makefileAssignment

driver.c file2.c

driver.o file2.o

matt@trout:makefileAssignment$ make clean

rm -fr makefileAssignment driver.o file1.o file2.o

matt@trout:makefileAssignment$ ls

Makefile driver.c file2.c

build file1.c headers.h

matt@trout:makefileAssignment$