

COSC 3750, Spring 2020
Linux Programming
Homework 3, Using Make

1 Intro

This time your job is to create a Makefile. I am giving you four (4) C source code files and three (3) C header files. You **will not** modify these files. This assignment is worth 50 points. The files are available on the homework assignment page. Just copy them into a clean directory.

2 Your job

Create a Makefile to compile the supplied files. This Makefile will conform to the following:

1. It will be named Makefile. Note the capital M.
2. It will have a heading that follows the format of specified in the Programming Style guide on the course homepage. Note that in Makefiles, the `#` is the comment start character and comments go to the end of the line. The long lines of `#`'s at the beginning and end in the example are just make this comment stand out.

```
#####  
# filename  
# Student's name  
# COSC 3750 and semester  
# Assignment  
# Date last modified  
#  
# Some discussion of what this file is about. Does not have to be  
# long winded.  
#####
```

3. You will specify reasonable values for the `CC` and `CFLAGS` variables. `CFLAGS` will contain at least an option for including `gdb` debugging information.
4. You will declare, AND USE, a variable **OBJS** which will contain the list of object files that *make* will generate.

5. You will NOT use an “all” target.
6. You **will** have a “.PHONY” target with any and all *phony* targets listed. This shall be the first target.
7. You will have one target for each of the objects, **prompt.o**, **compute.o**, and **display.o**. Each object is dependent on its source file and its header file. That is **prompt.o** depends on **prompt.c** and **prompt.h**.
8. You will have one target for the executable, **approx**. It is dependent on its source file and the three object files.
9. You will have a target **tidy** which will remove the object files. Make sure that you use “/bin/rm -f”. This makes sure that real (non-aliased) version of *rm* is used and the -f is to keep it from having a problem if a file does not exist.
10. You will have a target **clean** which will remove the object files **and** the executable.
11. To test this, I will copy your Makefile into a directory with the other files and type *make* at the prompt. I should generate the executable with no errors. I should NOT have to type anything else or modify anything in the Makefile.
12. Add comments as reasonable to ensure that you and others understand what your file is doing, what your variables are for (CC and CFLAGS are obvious, don’t comment), those kinds of things. Especially comment anything that is unusual, strange, odd, difficult, etc.

3 Submission

Submit online on WyoCourses are usual. You should only submit ONE file, **Makefile**.