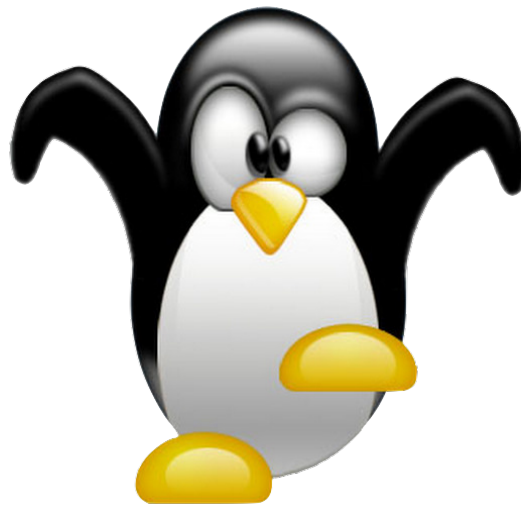


Managing Linux with the Embedded Perspective

Exercise 3(GNU Make)

Joseph Hotchkiss

September 1, 2024



Introduction

Exercise 2 required you to write a C program with multiple files. You were asked to compile and link the code with from the terminal and you were supposed to use the different compiler flags and investigate the individual steps of building a C program.

Basic Make

Create a basic make file that will build your code. The makefile that you create should be specific to your program. Everything should be written explicitly. There should be a rule for each step. There is no need for C or linker flags in this file. Each file should be compiled with its own recipe. And then the executable should be created with a separate recipe. Also create a target and corresponding recipe for cleaning your directory of unnecessary files (.o files).

Note: No wildcards, tricks or variables should be used here

Compact Makefile

Here you should create a makefile that is as compact as possible. You should use variables and wildcards wherever possible. It should make use of all of the compiler and linker flags that were required from the last exercise and any additional flags that you would like to be there.

Feel free to experiment with make. Please comment your makefiles to show that you have an understanding of what you have done. Comments begin with a '#' symbol.

Things that should be contained in variables:

- compiler
- source files
- compiler flags
- linker flags (if you use some)
- object files
- anything else you can think of

In addition you should use any tricks you can to make this makefile compact, such as wildcards and automatic variables.

Please note that we are only brushing the surface of what is possible with Make, but hopefully this gives you an idea and you will be able to see the use of this tool.

Return

Return your two make files to oma once it is back in service.