

A Very, Very Simple Makefile Tutorial

Makefiles are a simple way to organize code compilation. This tutorial gives you a very, very simple example of creating and using makefiles.

In Lab 5, you would compile the two .c files by executing the following command:

```
cc -o lab5 lab5.c lab5main.c
```

The above command compiles the two .c files and names the executable lab5. Without a makefile, the typical approach to the test/modify/debug cycle is to use the up arrow in a terminal to go back to your last compile command so you don't have to type it each time, especially once you've added a few more .c files to the mix.

If you lose the compile command or switch computers you have to retype it from scratch, which is inefficient at best. Makefiles can help to solve this inefficiency problem.

(Another inefficiency is that if you are only making changes to one .c file, you have to re-compile both every time without a makefile. Makefiles can help avoid this inefficiency, but we do not discuss the solution in this tutorial.)

Following is a makefile for Lab 5 programs:

```
lab5: lab5.c lab5main.c
    cc -o lab5 lab5.c lab5main.c
```

If you put this rule into a file called `Makefile` or `makefile` and then type `make` on the command line it will execute the compile command as you have written it in the makefile. Note that `make` with no arguments executes the first rule in the file. Furthermore, by putting the list of files on which the command depends on the first line after the `:`, `make` knows that the rule `lab5` needs to be executed if any of those files change. This way you can avoid using the up arrow repeatedly, looking for your last compile command.

After saving `Makefile` (or `makefile`), issue the following commands:

```
% make      # to compile both files
% lab5      # to run your C program lab5
```

One very important thing to note is that there is a tab before the `cc` command in the makefile. **There must be a tab at the beginning of any command**, and `make` will not be happy if it is not there.

Adapted from “A Simple Makefile Tutorial” written by Prof. Bruce A. Maxwell,
<http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/>