Compiling Overview

Once you have your code written, you need to compile it in order to be able to run it. Compiling a program is the act of translating the human-readable code that a programmer wrote (called “source code”) into a machine-executable format. The compiler is the program that performs this process for you: it takes your source code as input, and writes out the actual executable file, which you can then run.

At its simplest, compiling your program is a matter of running the compiler and giving it a command line argument specifying the .c file with the source code for your program. There are many different C compilers, but we will be using gcc, which stands for “GNU Compiler Collection.” If you wrote your program in a file called myProgram.c, you could execute the command gcc myProgram.c to compile your code. Assuming that there are no errors in your code, the compiler would produce an executable program called a.out, which you could then execute by typing the command ./a.out. However, if you try this on any of the C code you wrote in the previous chapters, you will encounter errors—we need to add a bit more to our code (which we will see shortly) to make it compile.

If you are not familiar with using a UNIX/Linux command line interface, you might want to review the UNIX readings in the previous module for more information about using the command line and command line arguments. We will go into more depth about command line arguments in Course 4. Typically, you will invoke the compiler with more arguments than just the name of one source file. We will discuss those options after we cover some more about the compilation process.

✓

Completed