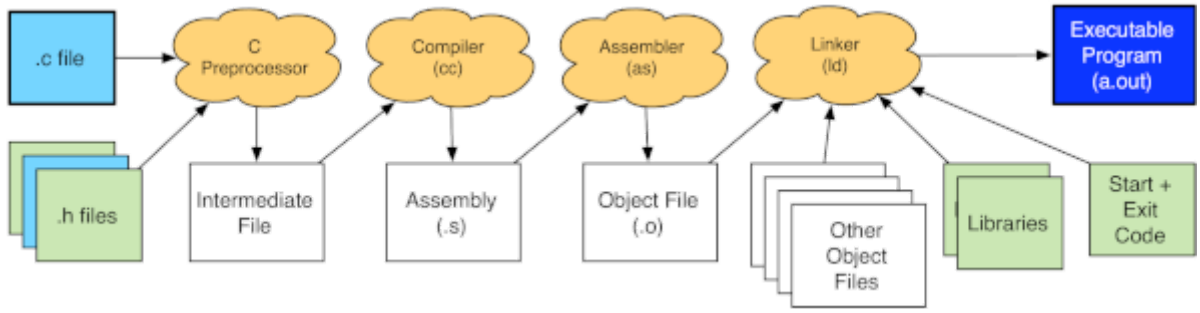
Compilation Process

coursera.org/learn/writing-running-fixing-code/supplement/AWnUk/compilation-process

When you are programming, you will use the compiler all the time, so it is useful to have a bit of an understanding of what it does at a high level. The inner workings and details of the compiler are quite complex and require an in-depth understanding of many areas of computer science and engineering, so we will not go into those. Instead, we will just cover the parts relevant to day-to-day program writing.



This figure illustrates a high-level view of the process that gcc takes to compile a program. Light blue boxes represent code you have written; light green boxes represent built-in parts of C.

The figure above shows a high-level overview of the process that gcc goes through to compile the code. In this picture, the light blue boxes represent code you have written, while the light green boxes represent built-in parts of C. The orange clouds indicate steps of this process (each is a separate program, but gcc invokes these programs for you), and the white boxes represent intermediate files that gcc generates to pass information from one stage to the next. The dark blue box in the upper right represents the final executable —the program that you can run to make your computer do whatever the program tells it to do.

✓

Completed