

The Actual Compiler

 coursera.org/learn/writing-running-fixing-code/supplement/uaXy3/the-actual-compiler

The output of the preprocessor is stored in a temporary file, and passed to the actual compiler. At first, it may seem a bit confusing that one of the steps of the compilation process is the actual compiler. However, we often refer to complex processes by their eponymous step. For example, if you say “I am going to bake a cake,” even though only part of the process involves actually baking the cake (cooking the cake batter in the oven), we still understand that you will go through the entire process: finding a recipe, getting the ingredients, mixing the batter, baking the cake, letting it cool, and icing it.

The compiler reads the pre-processed source code—which has all the specified files included and all macro definitions expanded—and translates it into assembly. Assembly is the lowest level type of human readable code. In assembly, each statement corresponds to one machine instruction. Machine instructions correspond to very simple operations such as adding two numbers or moving a value to or from memory. While humans can program in assembly (sometimes with good reason), that is not our focus here—we just want to understand enough of the compilation process to know what is going on when we compile our programs.



Completed
