

15 Writing Large Programs

Around computers it is difficult to find the correct unit of time to measure progress. Some cathedrals took a century to complete. Can you imagine the grandeur and scope of a program that would take as long?

Although some C programs are small enough to be put in a single file, most aren't. Programs that consist of more than one file are the rule rather than the exception. In this chapter, we'll see that a typical program consists of several source files and usually some header files as well. Source files contain definitions of functions and external variables; header files contain information to be shared among source files. Section 15.1 discusses source files, while Section 15.2 covers header files. Section 15.3 describes how to divide a program into source files and header files. Section 15.4 then shows how to “build” (compile and link) a program that consists of more than one file, and how to “rebuild” a program after part of it has been changed.

15.1 Source Files

Up to this point, we've assumed that a C program consists of a single file. In fact, a program may be divided among any number of *source files*. By convention, source files have the extension `.c`. Each source file contains part of the program, primarily definitions of functions and variables. One source file must contain a function named `main`, which serves as the starting point for the program.

For example, suppose that we want to write a simple calculator program that evaluates integer expressions entered in Reverse Polish notation (RPN), in which operators follow operands. If the user enters an expression such as

```
30 5 - 7 *
```

we want the program to print its value (175, in this case). Evaluating an RPN expression is easy if we have the program read the operands and operators, one by one, using a stack to keep track of intermediate results. If the program reads a