20 Low-Level Programming

A programming language is low level when its programs require attention to the irrelevant.

Previous chapters have described C's high-level, machine-independent features. Although these features are adequate for many applications, some programs need to perform operations at the bit level. Bit manipulation and other low-level operations are especially useful for writing systems programs (including compilers and operating systems), encryption programs, graphics programs, and programs for which fast execution and/or efficient use of space is critical.

Section 20.1 covers C's bitwise operators, which provide easy access to both individual bits and bit-fields. Section 20.2 then shows how to declare structures that contain bit-fields. Finally, Section 20.3 describes how certain ordinary C features (type definitions, unions, and pointers) can help in writing low-level programs.

Some of the techniques described in this chapter depend on knowledge of how data is stored in memory, which can vary depending on the machine and the compiler. Relying on these techniques will most likely make a program nonportable, so it's best to avoid them unless absolutely necessary. If you do need them, try to limit their use to certain modules in your program; don't spread them around. And, above all, be sure to document what you're doing!

20.1 Bitwise Operators

C provides six ***bitwise operators***, which operate on integer data at the bit level. We'll discuss the two bitwise shift operators first, followed by the four other bitwise operators (bitwise complement, bitwise *and*, bitwise exclusive *or*, and bitwise inclusive *or*).