22 Input/Output

In man-machine symbiosis, it is man who must adjust: The machines can't.

C's input/output library is the biggest and most important part of the standard library. As befits its lofty status, we'll devote an entire chapter (the longest in the book) to the `<stdio.h>` header, the primary repository of input/output functions.

We've been using `<stdio.h>` since Chapter 2, and we have experience with the `printf`, `scanf`, `putchar`, `getchar`, `puts`, and `gets` functions. This chapter provides more information about these six functions, as well as introducing a host of new functions, most of which deal with files. Fortunately, many of the new functions are closely related to functions with which we're already acquainted. `fprintf`, for instance, is the "file version" of the `printf` function.

We'll start the chapter with a discussion of some basic issues: the stream concept, the `FILE` type, input and output redirection, and the difference between text files and binary files (Section 22.1). We'll then turn to functions that are designed specifically for use with files, including functions that open and close files (Section 22.2). After covering `printf`, `scanf`, and related functions for "formatted" input/output (Section 22.3), we'll look at functions that read and write unformatted data:

- `getc`, `putc`, and related functions, which read and write one *character* at a time (Section 22.4).
- `gets`, `puts`, and related functions, which read and write one *line* at a time (Section 22.5).
- `fread` and `fwrite`, which read and write *blocks* of data (Section 22.6).

Section 22.7 then shows how to perform random access operations on files. Finally, Section 22.8 describes the `sprintf`, `snprintf`, and `sscanf` functions, variants of `printf` and `scanf` that write to a string or read from a string.

This chapter covers all but eight of the functions in `<stdio.h>`. One of these eight, the `perror` function, is closely related to the `<errno.h>` header, so