copying program, on the other hand, can't assume that the file to be copied is a text file. If it does, binary files containing an end-of-file character won't be copied completely. When we can't say for sure whether a file is text or binary, it's safer to assume that it's binary.

22.2 File Operations

Simplicity is one of the attractions of input and output redirection; there's no need to open a file, close a file, or perform any other explicit file operations. Unfortunately, redirection is too limited for many applications. When a program relies on redirection, it has no control over its files; it doesn't even know their names. Worse still, redirection doesn't help if the program needs to read from two files or write to two files at the same time.

When redirection isn't enough, we'll end up using the file operations that `<stdio.h>` provides. In this section, we'll explore these operations, which include opening a file, closing a file, changing the way a file is buffered, deleting a file, and renaming a file.

Opening a File

```
FILE *fopen(const char * restrict filename,
            const char * restrict mode);
```

fopen

Opening a file for use as a stream requires a call of the `fopen` function. `fopen`'s first argument is a string containing the name of the file to be opened. (A "file name" may include information about the file's location, such as a drive specifier or path.) The second argument is a "mode string" that specifies what operations we intend to perform on the file. The string `"r"`, for instance, indicates that data will be read from the file, but none will be written to it.

restrict keyword ➤ 17.8

C99

Note that `restrict` appears twice in the prototype for the `fopen` function. `restrict`, which is a C99 keyword, indicates that `filename` and `mode` should point to strings that don't share memory locations. The C89 prototype for `fopen` doesn't contain `restrict` but is otherwise identical. `restrict` has no effect on the behavior of `fopen`, so it can usually just be ignored. In this and subsequent chapters, I'll italicize *restrict* as a reminder that it's a C99 feature.

escape sequences ➤ 7.3

Windows programmers: Be careful when the file name in a call of `fopen` includes the \ character, since C treats \ as the beginning of an escape sequence. The call

```
fopen("c:\project\test1.dat", "r")
```

will fail, because the compiler treats `\t` as a character escape. (`\p` isn't a valid character escape, but it looks like one. The C standard states that its meaning is