```
while (buffer not full) {
  wait for input;
  buffer[i] = *p;
  if (buffer[i++] == '\n')
    break;
}
```

A sophisticated compiler might notice that this loop changes neither p nor *p, so it could optimize the program by altering it so that *p is fetched just once:

```
store *p in a register;
while (buffer not full) {
  wait for input;
  buffer[i] = value stored in register;
  if (buffer[i++] == '\n')
    break;
}
```

The optimized program will fill the buffer with many copies of the same character—not exactly what we had in mind. Declaring that p points to volatile data avoids this problem by telling the compiler that *p must be fetched from memory each time it's needed.

Q & A

Q: What do you mean by saying that the & and | operators sometimes produce the same results as the && and | | operators, but not always? [p. 511]

A: Let's compare `i & j` with `i && j` (similar remarks apply to | and | |). As long as `i` and `j` have the value 0 or 1 (in any combination), the two expressions will have the same value. However, if `i` and `j` should have other values, the expressions may not always match. If `i` is 1 and `j` is 2, for example, then `i & j` has the value 0 (`i` and `j` have no corresponding 1 bits), while `i && j` has the value 1. If `i` is 3 and `j` is 2, then `i & j` has the value 2, while `i && j` has the value 1.

Side effects are another difference. Evaluating `i & j++` *always* increments `j` as a side effect, whereas evaluating `i && j++` *sometimes* increments `j`.

Q: Who cares how DOS stores file dates? Isn't DOS dead? [p. 516]

A: For the most part, yes. However, there are still plenty of files created years ago whose dates are stored in the DOS format. In any event, DOS file dates are a good example of how bit-fields are used.

Q: Where do the terms "big-endian" and "little-endian" come from? [p. 520]

A: In Jonathan Swift's novel *Gulliver's Travels*, the fictional islands of Lilliput and Blefuscu are perpetually at odds over whether to open boiled eggs on the big end or the little end. The choice is arbitrary, of course, just like the order of bytes in a data item.