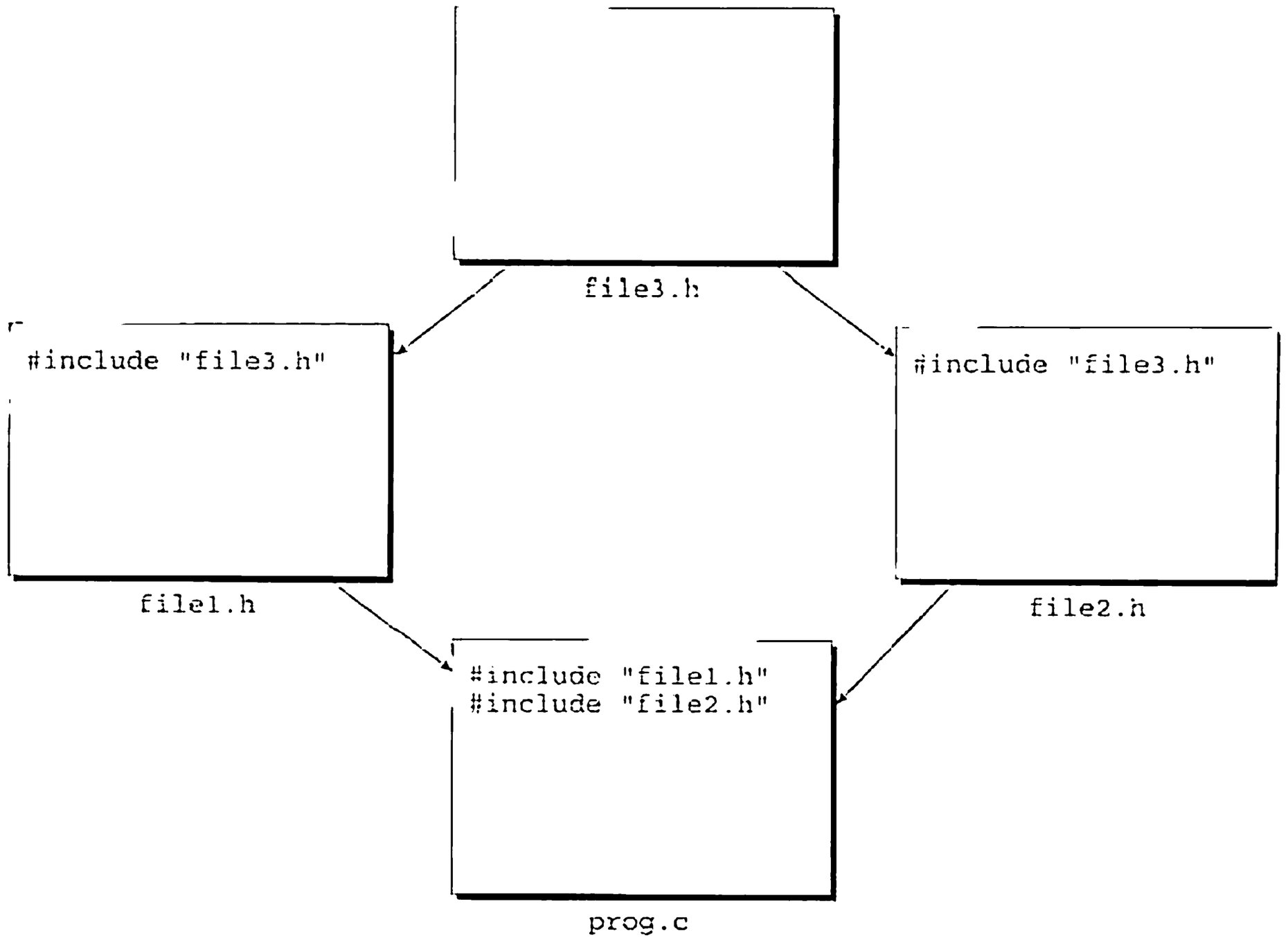

To protect a header file, we'll enclose the contents of the file in an `#ifndef`-`#endif` pair. For example, the `boolean.h` file could be protected in the following way:

```
#ifndef BOOLEAN_H
#define BOOLEAN_H

#define TRUE 1
#define FALSE 0
typedef int Bool;

#endif
```

When this file is included the first time, the `BOOLEAN_H` macro won't be defined, so the preprocessor will allow the lines between `#ifndef` and `#endif` to stay. But if the file should be included a second time, the preprocessor will remove the lines between `#ifndef` and `#endif`.

The name of the macro (`BOOLEAN_H`) doesn't really matter. However, making it resemble the name of the header file is a good way to avoid conflicts with other macros. Since we can't name the macro `BOOLEAN.H` (identifiers can't contain periods), a name such as `BOOLEAN_H` is a good alternative.

`#error` Directives in Header Files

`#error` directives ➤ *14.5*

`#error` directives are often put in header files to check for conditions under which the header file shouldn't be included. For example, suppose that a header