

Next: [Syntactic Conventions](#), Previous: [Formatting](#), Up: [Writing C](#) [[Contents](#)][[Index](#)]

5.2 Commenting Your Work

Every program should start with a comment saying briefly what it is for. Example: ‘fmt - filter for simple filling of text’. This comment should be at the top of the source file containing the ‘main’ function of the program.

Also, please write a brief comment at the start of each source file, with the file name and a line or two about the overall purpose of the file.

Please write the comments in a GNU program in English, because English is the one language that nearly all programmers in all countries can read. If you do not write English well, please write comments in English as well as you can, then ask other people to help rewrite them. If you can’t write comments in English, please find someone to work with you and translate your comments into English.

Please put a comment on each function saying what the function does, what sorts of arguments it gets, and what the possible values of arguments mean and are used for. It is not necessary to duplicate in words the meaning of the C argument declarations, if a C type is being used in its customary fashion. If there is anything nonstandard about its use (such as an argument of type `char *` which is really the address of the second character of a string, not the first), or any possible values that would not work the way one would expect (such as, that strings containing newlines are not guaranteed to work), be sure to say so.

Also explain the significance of the return value, if there is one.

Please put two spaces after the end of a sentence in your comments, so that the Emacs sentence commands will work. Also, please write complete sentences and capitalize the first word. If a lower-case identifier comes at the beginning of a sentence, don’t capitalize it! Changing the spelling makes it a different identifier. If you don’t like starting a sentence with a lower case letter, write the sentence differently (e.g., “The identifier lower-case is ...”).

The comment on a function is much clearer if you use the argument names to speak about the argument values. The variable name itself should be lower case, but write it in upper case when you are speaking about the value rather than the variable itself. Thus, “the inode number `NODE_NUM`” rather than “an inode”.

There is usually no purpose in restating the name of the function in the comment before it, because readers can see that for themselves. There might be an exception when the comment is so long that the function itself would be off the bottom of the screen.

There should be a comment on each static variable as well, like this:

```
/* Nonzero means truncate lines in the display;
   zero means continue them. */
int truncate_lines;
```

Every ‘`#endif`’ should have a comment, except in the case of short conditionals (just a few lines) that are not nested. The comment should state the condition of the conditional that is ending, *including its sense*. ‘`#else`’

should have a comment describing the condition *and sense* of the code that follows. For example:

```
#ifdef foo
...
#else /* not foo */
...
#endif /* not foo */
#ifdef foo
...
#endif /* foo */
```

but, by contrast, write the comments this way for a ‘#ifndef’:

```
#ifndef foo
...
#else /* foo */
...
#endif /* foo */
#ifndef foo
...
#endif /* not foo */
```

Next: [Syntactic Conventions](#), Previous: [Formatting](#), Up: [Writing C](#) [[Contents](#)][[Index](#)]