

**NAME**

style – Plan 9 coding conventions for C

**DESCRIPTION**

Plan 9 C code has its own conventions. You would do well to follow them. Here are a few (and this is not an exhaustive list):

- don't use `//` comments; some old Plan 9 code does, but we're converting it as we touch it. We do sometimes use `//` to comment-out a few lines of code.
- avoid **gotos**.
- no tabs expanded to spaces.
- surround a binary operator (particular a low precedence one) with spaces; don't try to write the most compact code possible but rather the most readable.
- parenthesize expressions involving arithmetic and bit-wise operators; otherwise don't parenthesize heavily (e.g., as in Pascal).
- no white space before opening braces.
- no white space after the keywords **if**, **for**, **while**, etc.
- no braces around single-line blocks (e.g., **if**, **for**, and **while** bodies).
- integer-valued functions return -1 on error, 0 or positive on success.
- functions that return errors should set *errstr*(2).
- variable and function names are all lowercase, with no underscores.
- **enum** or **#defined** constants should be Uppercase (or UPPERCASE).
- **struct** tags are Uppercase, with matching **typedefs**.
- automatic variables (local variables inside a function) are never initialized at declaration.
- follow the standard idioms: use `x < 0` not `0 > x`, etc.
- don't write **!strcmp** (nor **!memcmp**, etc.) nor **if(memcmp(a, b, c))**; always explicitly compare the result of string or memory comparison with zero using a relational operator.

Ultimately, the goal is to write code that fits in with the other code around it and the system as a whole. If the file you are editing already deviates from these guidelines, do what it does. After you edit a file, a reader should not be able to tell just from coding style which parts you worked on.

**COMMENTS**

If your code is readable, you shouldn't need many comments. A line or two comment above a function explaining what it does is always welcome.

Comment any code you find yourself wondering about for more than 2 seconds, even if it's to say that you don't understand what's going on. Explain why.

Don't use commenting as an excuse for writing confusing code. Rewrite the code to make it clear.

**EFFICIENCY**

Do the simple thing. Don't optimize unless you've measured the code and it is too slow. Fix the data structures and the algorithms instead of going for little 5% tunings.

**SEE ALSO**

"Notes on Programming in C", Rob Pike,  
<http://www.literateprogramming.com/pikestyle.pdf>

**BUGS**

Some programs use very different styles, for example, *rc*.

Some programs and programmers diverge from the above rules due to habits formed long before these rules. Notably, some programs have a single space after a keyword and before an opening brace, and some

initialize automatic variables at declaration.