

C Programming

- Four Rules for Better C Programming
 - Use strict ANSI C constructs
 - Never use global variables
 - Fix all compiler warnings
 - Use prototypes for all subroutines

C Programming

- Use strict ANSI C constructs

All C compilers allow for enforcement of strict ANSI C. You should use these options at all times. With the gcc compiler under the Linux operating system, the compiler flags are

`gcc -ansi -Wall -pedantic`

These flag settings will insure that you receive warnings for any poor programming constructs in your code.

C Programing

- Never use global variables

Global variables should never be used. There are exceptions to this rule, but you will never face a situation where they are truly needed. Global variables will lead to “spagetti code”, that can not be understood or debugged.

C Programming

- Fix all compiler warnings

Any warning from the compiler means that you have an error in your program or are using poor programming style. Either problem is serious and should be corrected. Typically, warnings are generated when variables are not properly retyped. The only exception to this rule is when you dynamically allocate memory. In this case, you may receive a warning about the byte location.

C Programming

- Use prototypes for all subroutines

Every subroutine you write should have a prototype that defines its argument types. This prototype should generally be contained in a single header file with a name such as *defs.h*.