

Re: Compile the 'sock' program in the book 'TCP/IP Illustrated Vol.1'

Source: <http://unix.derkeiler.com/Newsgroups/comp.unix.programmer/2005-11/0718.html>

From: Michael Wojcik (*mwojcik_at_newsguy.com*)

Date: 11/21/05

Date: 21 Nov 2005 05:09:05 GMT

In article <87lkzjpbsw.fsf@narke.yellow.line>, Steven Woody <anti-spam.narkewoody@gmail.com.dont-post-to> writes:

>
> *i am running Linux and want to compile and use the 'sock' program comes with*
> *Stevens's book 'TCP/IP Illustrated Vol.1'. but the code simply can't pass the*
> *compiler with following error,*
>
> *| bash-3.00\$ make*
> *| gcc -ansi -Wall -Dsun -D__STDC__=0 -c -o loop.o loop.c*
> *| <command line>:5:1: warning: "__STDC__" redefined*

Why are you defining `__STDC__` at all, much less to 0? `__STDC__` should always be defined by the implementation. If this is from the makefile that you got with the source for sock, be aware that `_TCP/IP Illustrated_v1`, while an excellent book, is more than ten years old, and the makefile might require some reworking for newer systems. (Though even in 1994 defining `__STDC__` explicitly was almost certainly wrong.)

> *| loop.c: In function `loop':*
> *| loop.c:117: error: `caddr_t' undeclared (first use in this function)*
> ...
>
> *in fact, in the Linux system, there is a sys/types.h which defined the caddr_t*
> *type and this file seems have been already included in the source code, so i*
> *don't know why the type still can not be recognized.*

Because it hasn't been defined. Including `sys/types.h` very likely doesn't guarantee that a particular typedef in it has been processed, because in most implementations `sys/types.h` is full of conditional-inclusion directives (`#if` and friends).

In the Linux `sys/types.h` I'm looking at now, `caddr_t` is only defined if the macro `__USE_BSD` is defined. Note that this is an identifier reserved to the implementation, so you shouldn't be defining it

