

Hey guys!

Here I post what Linus Torvalds has to say about -Wshadow (from <http://kerneltrap.org/node/7434>). Please consider removing this warning.

Quote:

On Wed, 29 Nov 2006, Jesper Juhl wrote:

>

> I would venture that "-Wshadow" is another one of those.

I'd agree, except for the fact that gcc does a horribly bad job of -Wshadow, making it (again) totally unusable.

For example, it's often entirely interesting to hear about local variables that shadow each other. No question about it.

HOWEVER. It's not really interesting to hear about a local variable that happens to have a common name that is also shared by a extern function.

There just isn't any room for confusion, and it's actually not even that unusual - I tried using -Wshadow on real programs, and it was just horribly irritating.

In the kernel, we had obvious things like local use of "jiffies" that just make total sense in a small inline function, and the fact that there happens to be an extern declaration for "jiffies" just isn't very interesting.

Similarly, with nested macro expansion, even the "local variable shadows another local variable" case - that looks like it should have an obvious warning on the face of it - really isn't always necessarily that interesting after all. Maybe it is a bug, maybe it isn't, but it's no longer obviously bogus any more.

So I'm not convinced about the usefulness of "-Wshadow". ESPECIALLY the way that gcc implements it, it's almost totally useless in real life.

For example, I tried it on "git" one time, and this is a perfect example of why "-Wshadow" is totally broken:

```
diff-delta.c: In function 'create_delta_index':  
diff-delta.c:142: warning: declaration of 'index' shadows a global declaration
```

(and there's a lot of those). If I'm not allowed to use "index" as a local variable and include <string.h> at the same time, something is simply **SERIOUSLY WRONG** with the warning.

So the fact is, the C language has scoping rules for a reason. Can you

screw yourself by usign them badly? Sure. But that does NOT mean that the same name in different scopes is a bad thing that should be warned about.

If I wanted a language that didn't allow me to do anything wrong, I'd be using Pascal. As it is, it turns out that things that "look" wrong on a local level are often not wrong after all.

Linus
