
char unsigned char and packed structs

Posted by vipps...@gmail.com - 2009/07/17 23:13

Hello comp.std.c Why isn't there a way to tell the compiler to pack the structure? (pack: no padding bytes, sizeof struct equal to sum of sizeof members) You might argue this is provided by most implementations and not really an issue, but way I see it, there's no reason not to have such feature, unless I'm mistaken about the following: (excuse the terminology) 1) an object is really just multiple of unsigned char. (that's what sizeof reports) 2) it's possible to treat any unsigned char m as *(T*)m

=====

char unsigned char and packed structs

Posted by Keith Thompson - 2009/07/17 23:13

Hello comp.std.c Why isn't there a way to tell the compiler to pack the structure? (pack: no padding bytes, sizeof struct equal to sum of sizeof members) You might argue this is provided by most implementations and not really an issue, but way I see it, there's no reason not to have such feature, unless I'm mistaken about the following: (excuse the terminology) 1) an object is really just multiple of unsigned char. (that's what sizeof reports) 2) it's possible to treat any unsigned char m as *(T*)m

=====

char unsigned char and packed structs

Posted by vipps...@gmail.com - 2009/07/17 23:13

vipps...@gmail.com writes: 1) an object is really just multiple of unsigned char. (that's what sizeof reports) 2) it's possible to treat any unsigned char m as *(T*)m

=====

char unsigned char and packed structs

Posted by Keith Thompson - 2009/07/17 23:13

vipps...@gmail.com writes: 1) an object is really just multiple of unsigned char. (that's what sizeof reports) 2) it's possible to treat any unsigned char m as *(T*)m

=====

char unsigned char and packed structs

Posted by vipps...@gmail.com - 2009/07/17 23:13

This could certainly be defined, but it would be a fairly major change to the language

=====

char unsigned char and packed structs

Posted by Jack Klein - 2009/07/17 23:13

vipps...@gmail.com writes: 1) an object is really just multiple of unsigned char. (that's what sizeof reports) 2) it's possible to treat any unsigned char m as *(T*)m

=====