Size of struct with a single element

Ask Question

Asked 12 years, 11 months ago Modified 12 years, 11 months ago Viewed 3k times

18

Given

```
struct S {
  SomeType single_element_in_the_struct;
};
```

Is it always true that

```
sizeof(struct S) == sizeof(SomeType)
```

Or it may be implementation dependent?

c++ c

Share Improve this question Follow

edited Aug 27, 2010 at 14:30
Puppy
145k 37 255 465

asked Aug 27, 2010 at 14:17
user396672
3,106 1 20 31

Add a comment

3 Answers

Sorted by: Highest score (default)

14

This will usually be the case, but it's not guaranteed.

Any struct may have unnamed padding bytes at the end of the struct, but these are usually used for alignment purposes, which isn't a concern if you only have a single element.

Share Improve this answer Follow

answered Aug 27, 2010 at 14:20
James McNellis
348k 75 913 977

3 *which isn't a concern if you only have a single element* Actually handling a single element in a struct may still be padded for alignment / performance reasons. Typically example is a `char array[6]` on a 32-bit system, may pad with 2 additional bytes to be a whole multiple of 4-bytes (32-bits). – mctylr Aug 27, 2010 at 14:48

1 @mctylr: I'd be a bit surprised if that behavior was typical. There can be no padding between elements of an array, so a `char[6]` will have a size of six bytes and a `char[6][2]` will have a size of 12 bytes. I don't see the benefit of aligning a type `struct S { char c[6]; };` differently than the element of type `char[6]` that it contains. – James McNellis Aug 27, 2010 at 15:04

1 I meant treat it as `struct S = {char array[6]; char pad[2];}` Since the a 32-bit CPU is going to access 8 bytes anyhow, if `struct S` is itself within a second array `struct S aS[]` the elements of the array are then aligned on alignment boundaries. I wouldn't expect it to be common, but possible. I've seen people assume no such padding when directly writing structs via fwrite / write to a binary file or network socket, and it can and does break. – mctylr Sep 3, 2010 at 15:01

Additional case: Consider a word-addressed machine where sizeof(char*) > sizeof(int*). (The char pointer has to have additional bits to select with a word.). The standard guarantees that all pointers to struct have the same size, so a struct will typically always be word-aligned (you could choose to say all struct pointers will be unaligned, but that was typically not the choice on such machines). If all structs are word aligned, sizeof(struct S { char a;}) will be greater than sizeof(char). – Martin Bonner supports Monica Oct 10, 2017 at 9:28

Add a comment

13

It does not have to be equal, due to structure padding.

section 6.7.2.1 in the C99 standard states that "There may be unnamed padding within a structure object, but not at its beginning".

This is refered to as structure padding. Paddings may be added to make sure that the structure is properly aligned in memory. The exakt size of a structure can change if you change the order of its members.

Share Improve this answer Follow

edited Aug 27, 2010 at 14:27

answered Aug 27, 2010 at 14:22
Silfverstrom
28.2k 6 45 57

1 Or, two paragraphs later, "There may be unnamed padding at the end of a structure or union." – James McNellis Aug 27, 2010 at 14:27

That's not particularly useful, since the OP tagged both C++ and C. – Puppy Aug 27, 2010 at 14:31

3 But does padding come into play if there's only one member of the struct? That's the OP's question, and so far I haven't found anything definitive in n1256. For example, per gcc, the size of a struct containing a single 3-element array of char is 3; no padding. Add an int following the array, and the size of the array is 8, so there's one byte of padding between the elements. – John Bode Aug 27, 2010 at 19:12

Add a comment

1

It depends on the packing of your compiler. Usually the size of a structure is divisible by the word-length of your system (e.g. 4 byte == 32 bit).

So you will often have `sizeof(struct S) > sizeof(SomeType)`

For most compilers you can modify the packing size using compiler pragmas. If you set `#pragma pack(1)` then the sizes should be equal.

Share Improve this answer Follow

answered Aug 27, 2010 at 14:26
SebastianK
3,572 3 30 48

1 usually it doesn't depend on packing. Alignment of a structure usually is calculated as a strongest alignment of structure members. If all members are chars then alignment in most cases will be 1 (regardless to default packing). – user396672 Aug 27, 2010 at 14:41

Add a comment

Your Answer

Sign up or log in

Sign up using Google

Sign up using Facebook

Sign up using Email and Password

Post as a guest

Name

Email
Required, but never shown

Post Your Answer

By clicking "Post Your Answer", you agree to our terms of service and acknowledge that you have read and understand our privacy policy and code of conduct.

Not the answer you're looking for? Browse other questions tagged c++ c or ask your own question.

The Overflow Blog

Why everyone should be an AppSec specialist (Ep. 598)

Featured on Meta

Moderation strike: Results of negotiations

Our Design Vision for Stack Overflow and the Stack Exchange network

Temporary policy: Generative AI (e.g., ChatGPT) is banned

Preview of Search and Question-Asking Powered by GenAI

Collections: A New Feature for Collectives on Stack Overflow

Linked

12 Can wrapping a type in a struct cause additional padding?

13 Is sizeof(T) == sizeof(int)?

2 Spirit Qi First Parser

Related

3 size of struct and corresponding variable

3 Actual total size of struct's members

135 sizeof single struct member in C

1 size of struct-array

5 Size of a struct containing 1 Pointer

8 sizeof struct or variable in C and C++

2 How to calculate size of a struct in C?

5 Sizeof a struct in C

0 The size of a struct

0 What is the size of this struct?

Hot Network Questions

date - Can't Go Back More Than 115 Years or Can't Go 5879565 Years Into the Future

Does a "dark" sport make sense?

Is it morally wrong to enjoy art created by a heinous criminal?

Why is "diamond problem" a problem? Why doesn't the child simply call both parent's methods sequentially?

Can you write "... me's" (the possessive)?

Unable to ssh into headless raspberry pi?

How to deal with constant stress and uncertainty in academia?

Why is it justified to break a "bad" law because it's bad?

Iterate over all non-equivalent strings

How many police officers were killed in the police station in The Terminator?

Line numbers in \captionof

Is it possible to get a transparent cuboid that's opaque in bulk but does not have any faces?

Shift Braille down

What was the tipping point district for the 2020 presidential election under 2022 House lines?

Is the fingering strange in Saint-Saens piano version (from Liszt) of Danse macabre

Tabulary package does not print table with correct size

Symbol for car brakes

Why is drunk driving causing accident punished so much worse than just drunk driving?

What would be the primary weaknesses of massed archers?

Can I put all points in one ability with customized origins?

Why is the UKSA not called the Ministry of Space?

Sci-fi novel where main character is a computer tech that discovers they are on Earth using antenna shadows

Why does the EU believe that there's no need for an across the board harmonisation of member states' tax systems?

Does the Riemann hypothesis predict a bound for this prime-counting function?

Question feed

Site design / logo © 2023 Stack Exchange Inc; user contributions licensed under CC BY-SA rev 2023.8.14.43576