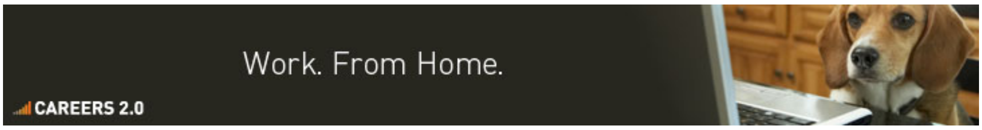
Structure padding



I am learning structure padding and packing in C. I have this doubt, as I have read padding will depend on architecture, so does it affect inter machine communication?, ie. if data created on one machine is getting read on other machine. How this problem is avoided in this scenario.

c

asked 9 hours ago
user2333014
14 1

3 Answers

Yes, you cannot send the binary data of a structure between platforms and expect it to look the same on the other side.

The way you solve it is you create a marshaller/demarshaller for your construct and pass it through on the way out of one system, and on the way in to the other system. This lets the compiler take care of the buffering for you on each system.

Each side knows how to take the data, as you've specified it will be sent, and deal with it for the local platform.

Platforms such as java handle this for you by creating serialization mechanisms for your classes. In C, you'll need to do this for yourself. How you do it depends on how you want to send your data. You could serialize to binary, XML, or anything else.

edited 8 hours ago

answered 9 hours ago
xaxxon
1,823 5 22

xdr is worth looking at. – anthony-arnold 8 hours ago

`#pragma pack` is supported by most compilers that I know of. This can allow the programmer to specify their desired padding method for structs.

http://msdn.microsoft.com/en-us/library/2e70t5y1%28v=vs.80%29.aspx

http://gcc.gnu.org/onlinedocs/gcc/Structure_002dPacking-Pragmas.html

http://clang.llvm.org/docs/UsersManual.html#microsoft-extensions

answered 9 hours ago
StilesCrisis
5,720 1 6 21

This is true, but grossly misleading in the context of communicating across a network or sharing files across platforms. – Ben Jackson 9 hours ago

Yes, you can change the structure padding. **This can be dangerous.** Some APIs require data to be aligned properly. Changing the padding can ruin this. – anthony-arnold 8 hours ago

And changing the structure padding isn't enough to ensure data is interpreted the same on different machines. There are many other matters, like how floats/doubles are represented, how signed integers are represented, what the byte order of integers are, what the size of a datatype is (`char` , `int` , `long` , `long long` ?), and heaps more. – anthony-arnold 8 hours ago

The question was specifically about padding. Obviously other issues are at play; endianness would be the next one in my mind. If you can ensure a matching structure pack, word size and endianness, for typical architectures at that point you are good. – StilesCrisis 8 hours ago

@StilesCrisis Yes, but the user asked about inter-machine communication. I thought they should know that structure padding is not the only problem to think about. – anthony-arnold 8 hours ago

show **1** more comment

It will only be affected if the code you have compiled for some other architecture uses a different padding scheme.

To help alleviate problems, I recommend that you pack structures with no padding. Where padding is required, use place-holders in (eg `char reserved[2]`). Also, don't use bitfields!! They are not portable.

You should also be aware of other architecture-related problems. Specifically endianness, and datatype sizes. If you need better portability, you may want to serialise and de-serialise a byte stream instead of casting it as a `struct` .

answered 9 hours ago
paddy
12.3k 1 6 28

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