

Re: strlen(), K+1: clarification

Source: <http://coding.derkeiler.com/Archive/General/comp.programming/2008-02/msg01651.html>

- *From:* Richard Heathfield <rjh@xxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 26 Feb 2008 19:20:07 +0000
-

Alan Morgan said:

<snip>

Heavens to Knuth, you rake a guy over the coals for brain farting and confusing && and ||

You have been spinozed. I didn't confuse them at all. I *misread* one of the answers in the multiple choice. Mr Nilges's continuous repetitions of falsehoods do not make those falsehoods true. I have made this distinction a number of times.

and you won't admit that using strlen(s) in a for loop condition is a bad idea.

Unlike the rest of us, Mr Nilges seems to think that reacting to one's own error by saying "oops, I was wrong" is a Bad Thing. Either that, or he still doesn't realise he was wrong, even though the whole matter has been explained to him in words of very few syllables, very many times. His capacity for learning is quite astonishing.

<snip>

As Richard unintentionally demonstrated, using memcpy means you need to check for overlap yourself to avoid a bug.

Wow. They should really document that somewhere...

Um, quite so. But of course I simply didn't consider the overlapping case, because it didn't occur to me that anyone would want to swap overlapping objects that did not overlap 100%. On reflection, I thought they might, so

Re: strlen(), K+1: clarification

I mentioned the fact. Everyone else seems to have reacted to this by saying "hmmm, no, actually that would be dumb because..." – except one person, of course.

And if you have to work with strings longer than cache length, it's probably a good idea to redo all strings as

```
struct TYPstring
{
  long intLength;
  char * strValue;
}
```

with an inspect function to make sure that the value is the length claimed. It is obscene that this has to be done, but that's C for you.

I don't see this as a major improvement.

Neither do I. But I /do/ see { size_t length; size_t capacity; char *data; } as a major improvement (given sufficient supporting functionality, of course).

Oh, sure, strlen() is much faster,

That's one advantage. But there are numerous others, if you make the string "stretchy" – resizeable, that is.

but how often do you call strlen()?

I don't know. I've never counted. :-)

<snip>

But as opposed to forgetting the behavior of memcpy and confusing && and || under pressure, this error is de minimis.

I've interviewed a lot of people for programming jobs. I'll excuse a lot of errors in an interview if the interviewee admits/notices the mistake and fixes it. Heck, I've had someone write Pascal code instead of C code (they noticed it, commented on the fact that they'd been using Delphi for some personal projects, erased the code, and

Re: strlen(), K+1: clarification

rewrote it in C). It happens.

Right. Note, however, that I didn't forget the behaviour of memcpy, and the claim that I did is utterly without foundation. What's more, I did not confuse && and || – I know perfectly well what they mean. I simply read three multiple choice answers at once, and mis-read one of them. There is a big difference.

--

Richard Heathfield <<http://www.cpax.org.uk>>

Email: rjh@http://www

Google users: <<http://www.cpax.org.uk/prg/writings/googly.php>>

"Usenet is a strange place" – dmr 29 July 1999

.