

# Re: why debug step by step, it s ok

---

*Source:* [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.c/2005-06/msg00575.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2005-06/msg00575.html)

---

- *From:* Keith Thompson <kst-u@xxxxxxx>
  - *Date:* Sat, 04 Jun 2005 18:03:53 GMT
- 

"Emmanuel Delahaye" <emdel@xxxxxxxxxxxxxxxx> writes:

> Robert Gamble wrote on 04/06/05 :  
>> I thought about clarifying that but decided not to assuming that the  
>> "without invoking undefined behavior" part was implied and trying not  
>> to be too pedantic as I know what Emmanuel meant ;)  
>  
> Ok, I rephrase it:  
>  
> "The C langage doesn't guarantee that a string literal is mutable".

Another way to put it is that you can't go wrong assuming that string literals are immutable.

That's not quite true if you're trying to figure out why code that modifies a string literal *\*seems\** to be working, but it's a good rule for writing new code.

—  
Keith Thompson (The\_Other\_Keith) kst-u@xxxxxxx <<http://www.ghoti.net/~kst>>  
San Diego Supercomputer Center <\*> <<http://users.sdsc.edu/~kst>>  
We must do something. This is something. Therefore, we must do this.

- 
- *References:*
    - ◆ [\*why debug step by step, it s ok\*](#)
      - ◇ *From:* baumann@pan
    - ◆ [\*Re: why debug step by step, it s ok\*](#)
      - ◇ *From:* Emmanuel Delahaye
    - ◆ [\*Re: why debug step by step, it s ok\*](#)
      - ◇ *From:* Robert Gamble
    - ◆ [\*Re: why debug step by step, it s ok\*](#)
      - ◇ *From:* Keith Thompson
    - ◆ [\*Re: why debug step by step, it s ok\*](#)
      - ◇ *From:* Robert Gamble
    - ◆ [\*Re: why debug step by step, it s ok\*](#)
      - ◇ *From:* Emmanuel Delahaye

Re: why debug step by step, it s ok

- Prev by Date: ***Re: Why compiler not generating any warning ?***
- Next by Date: ***Re: How to test whether strstr() returns a null pointer or not?***
- Previous by thread: ***Re: why debug step by step, it s ok***
- Next by thread: ***Re: why debug step by step, it s ok***
- Index(es):
  - ◆ ***Date***
  - ◆ ***Thread***