作者：李子天

链接：https://www.zhihu.com/question/28191923/answer/40028231

来源：知乎

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

C语言标准规定数组的值可以改变，而修改字符串字面量的值的结果是未定义的（详见分割线后，感谢 @Sunchy321 指正）。char s[LEN]和char \*s这两个s不是一种东西。这个问题编译器的具体实现一般是这样的：

char s[10] = "whatever";

也相当于

char s[10] = {'w','h','a','t','e','v','e','r','\0'};

会在stack分配10个字节给数组s，然后把"whatever"拷贝进去。stack是可读写的，所以你可以修改字符数组的值。

char \*s = "whatever";

会在stack分配sizeof(char \*)字节的空间给指针s，然后将s的值修改为"whatever"的地址，而这段地址一般位于只读数据段中。

在现代操作系统中，可以将一段内存空间设置为“读写数据”、“只读数据”等等多种属性，一般编译器会将"whatever"字面量放到像".rodata"这样的只读数据段中，修改只读段会触发 CPU 的保护机制(#GP)从而导致操作系统将程序干掉。

==========我是分割线==========

ISO/IEC 9899:1999 (E) , §6.4.5 String literals (P63L5):

If the program attempts to modify such an array, the behavior is undefined.

修改字符串字面量的行为是Undefined Behavior，也就是说具体实现是依赖编译器的，上文所举的例子只是x86下常见的情况，但并不是C语言标准所规定的。

比如，

char \*a = "WhatEver";

char \*b = "WhatEver";

编译器完全可以为节约空间只存储一个"WhatEver"，也就是实际上a = b，这种情况下，在没有段保护机制的系统中如果你

a[4] = "O";

很可能会导致 a 变为 "WhatOver" 的同时 b 也变为 "WhatOver"。