# **sizeof与strlen的区别**



[21aspnet](https://me.csdn.net/21aspnet" \t "https://blog.csdn.net/21aspnet/article/details/_blank) 2007-03-24 16:24:00 IMG_257 69274 IMG_258 收藏 60

分类专栏： [C基础](https://blog.csdn.net/21aspnet/category_851646.html" \t "https://blog.csdn.net/21aspnet/article/details/_blank) 文章标签： [存储](https://www.csdn.net/gather_2d/MtTaEg0sNDg1MTYtYmxvZwO0O0OO0O0O.html" \t "https://blog.csdn.net/21aspnet/article/details/_blank) [fun](https://so.csdn.net/so/search/s.do?q=fun&t=blog&o=vip&s=&l=&f=&viparticle=" \t "https://blog.csdn.net/21aspnet/article/details/_blank) [编译器](https://www.csdn.net/gather_27/MtTaEg0sMTA0NzctYmxvZwO0O0OO0O0O.html" \t "https://blog.csdn.net/21aspnet/article/details/_blank) [null](https://www.csdn.net/gather_20/MtTaEg0sMTk3MzItYmxvZwO0O0OO0O0O.html" \t "https://blog.csdn.net/21aspnet/article/details/_blank) [c++](https://www.csdn.net/gather_2c/Ntjacg0sNDk3Ny1ibG9n.html" \t "https://blog.csdn.net/21aspnet/article/details/_blank)

版权

[](https://blog.csdn.net/21aspnet/category_9867613.html" \t "https://blog.csdn.net/21aspnet/article/details/_blank)

**[精品专栏](https://blog.csdn.net/21aspnet/category_9867613.html" \t "https://blog.csdn.net/21aspnet/article/details/_blank)**

[精品高质量技术实战分享](https://blog.csdn.net/21aspnet/category_9867613.html" \t "https://blog.csdn.net/21aspnet/article/details/_blank)

[IMG_26021aspnet](https://blog.csdn.net/21aspnet/category_9867613.html" \t "https://blog.csdn.net/21aspnet/article/details/_blank)

**¥9.90**

订阅博主

char str[20]="0123456789";  
int   a=strlen(str); /\*a=10;strlen 计算字符串的长度，以\0'为字符串结束标记。  
int   b=sizeof(str); /\*b=20;sizeof 计算的则是分配的数组str[20] 所占的内存空间的大小，不受里面存储的内容影响  
============================================================================================

char \*str1="absde";  
char str2[]="absde";  
char str3[8]={'a',};  
char ss[] = "0123456789";

输出：

sizeof(str1)=4  
sizeof(str2)=6;  
sizeof(str3)=8;  
sizeof(ss)=11

首先说明一点，char类型占一个字节，所以sizeof(char)是1，这点要理解

str1是一个指针，只是指向了字符串"absde"而已。所以sizeof(str1)不是字符串占的空间也不是字符数组占的空间，而是一个字符型指针占的空间。所以sizeof(str1)=sizeof(char\*)=4，在C/C++中一个指针占4个字节

str2是一个字符型数组。C/C++规定，对于一个数组，返回这个数组占的总空间，所以sizeof(str2)取得的是字符串"absde"占的总空间。"absde"中，共有a b s d e \0六个字符，所以str2数组的长度是6，所以sizeof(str2)=6\*sizeof(char)=6

str3已经定义成了长度是8的数组，所以sizeof(str3)为8

str4和str2类似，'0' '1' ... '9'加上'\0'共11个字符，所以ss占的空间是8

总之，对于指针，sizeof操作符返回这个指针占的空间，一般是4个字节；而对于一个数组，sizeof返回这个数组所有元素占的总空间。char\*与char[]容易混淆，一定要分清，而且char\*="aaa"的写法现在不被提倡，应予以避免  
而strlen不区分是数组还是指针，就读到\0为止返回长度。而且strlen是不把\0计入字符串的长度的。  
============================================================================================  
一、sizeof  
    sizeof(...)是运算符，在头文件中typedef为unsigned int，其值在编译时即计算好了，参数可以是数组、指针、类型、对象、函数等。  
    它的功能是：获得保证能容纳实现所建立的最大对象的字节大小。  
    由于在编译时计算，因此sizeof不能用来返回动态分配的内存空间的大小。实际上，用sizeof来返回类型以及静态分配的对象、结构或数组所占的空间，返回值跟对象、结构、数组所存储的内容没有关系。  
    具体而言，当参数分别如下时，sizeof返回的值表示的含义如下：  
    数组——编译时分配的数组空间大小；  
    指针——存储该指针所用的空间大小（存储该指针的地址的长度，是长整型，应该为4）；  
    类型——该类型所占的空间大小；  
    对象——对象的实际占用空间大小；  
    函数——函数的返回类型所占的空间大小。函数的返回类型不能是void。  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*

二、strlen  
    strlen(...)是函数，要在运行时才能计算。参数必须是字符型指针（char\*）。当数组名作为参数传入时，实际上数组就退化成指针了。  
    它的功能是：返回字符串的长度。该字符串可能是自己定义的，也可能是内存中随机的，该函数实际完成的功能是从代表该字符串的第一个地址开始遍历，直到遇到结束符NULL。返回的长度大小不包括NULL。  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

三、举例：  
    eg1、char arr[10] = "What?";  
              int len\_one = strlen(arr);  
              int len\_two = sizeof(arr);  
              cout << len\_one << " and " << len\_two << endl;  
    输出结果为：5 and 10  
    点评：sizeof返回定义arr数组时，编译器为其分配的数组空间大小，不关心里面存了多少数据。strlen只关心存储的数据内容，不关心空间的大小和类型。

    eg2、char \* parr = new char[10];  
              int len\_one = strlen(parr);  
              int len\_two = sizeof(parr);  
              int len\_three = sizeof(\*parr);  
              cout << len\_one << " and " << len\_two << " and " << len\_three << endl;  
    输出结果：23 and 4 and 1  
    点评：第一个输出结果23实际上每次运行可能不一样，这取决于parr里面存了什么（从parr[0]开始知道遇到第一个NULL结束）；第二个结果实际上本意是想计算parr所指向的动态内存空间的大小，但是事与愿违，sizeof认为parr是个字符指针，因此返回的是该指针所占的空间（指针的存储用的是长整型，所以为4）;第三个结果，由于\*parr所代表的是parr所指的地址空间存放的字符，所以长度为1。  
\*\*\*\*\*\*\*\*\*\*\*\*

四、参考资料：  
Sizeof与Strlen的区别与联系(转）

1.sizeof操作符的结果类型是size\_t，它在头文件中typedef为unsigned int类型。  
该类型保证能容纳实现所建立的最大对象的字节大小。

2.sizeof是算符，strlen是函数。

3.sizeof可以用类型做参数，strlen只能用char\*做参数，且必须是以''\0''结尾的。  
sizeof还可以用函数做参数，比如：  
short f();  
printf("%d\n", sizeof(f()));  
输出的结果是sizeof(short)，即2。

4.数组做sizeof的参数不退化，传递给strlen就退化为指针了。

5.大部分编译程序 在编译的时候就把sizeof计算过了 是类型或是变量的长度这就是sizeof(x)可以用来定义数组维数的原因  
char str[20]="0123456789";  
int a=strlen(str); //a=10;  
int b=sizeof(str); //而b=20;

6.strlen的结果要在运行的时候才能计算出来，时用来计算字符串的长度，不是类型占内存的大小。

7.sizeof后如果是类型必须加括弧，如果是变量名可以不加括弧。这是因为sizeof是个操作符不是个函数。

8.当适用了于一个结构类型时或变量， sizeof 返回实际的大小，  
当适用一静态地空间数组， sizeof 归还全部数组的尺寸。  
sizeof 操作符不能返回动态地被分派了的数组或外部的数组的尺寸

9.数组作为参数传给函数时传的是指针而不是数组，传递的是数组的首地址，  
如：  
fun(char [8])  
fun(char [])  
都等价于 fun(char \*)  
在C++里参数传递数组永远都是传递指向数组首元素的指针，编译器不知道数组的大小  
如果想在函数内知道数组的大小， 需要这样做：  
进入函数后用memcpy拷贝出来，长度由另一个形参传进去  
fun(unsiged char \*p1, int len)  
{  
unsigned char\* buf = new unsigned char[len+1]  
memcpy(buf, p1, len);  
}

我们能常在用到 sizeof 和 strlen 的时候，通常是计算字符串数组的长度  
看了上面的详细解释，发现两者的使用还是有区别的，从这个例子可以看得很清楚：

char str[20]="0123456789";  
int a=strlen(str); //a=10; >>>> strlen 计算字符串的长度，以结束符 0x00 为字符串结束。  
int b=sizeof(str); //而b=20; >>>> sizeof 计算的则是分配的数组 str[20] 所占的内存空间的大小，不受里面存储的内容改变。

上面是对静态数组处理的结果，如果是对指针，结果就不一样了

char\* ss = "0123456789";  
sizeof(ss) 结果 4 ＝＝＝》ss是指向字符串常量的字符指针，sizeof 获得的是一个指针的之所占的空间,应该是

长整型的，所以是4  
sizeof(\*ss) 结果 1 ＝＝＝》\*ss是第一个字符 其实就是获得了字符串的第一位'0' 所占的内存空间，是char类

型的，占了 1 位

strlen(ss)= 10 >>>> 如果要获得这个字符串的长度，则一定要使用 strlen  
===========================================================================