<https://blog.csdn.net/okawari_richi/article/details/57411796>

strcpy()函数是C语言中的一个复制字符串的库函数，以下将详细解释说明一下：

### · 函数声明以及实现代码

**[cpp]** [view plain](https://blog.csdn.net/okawari_richi/article/details/57411796) [copy](https://blog.csdn.net/okawari_richi/article/details/57411796)

1. **char** \*strcpy(**char** \*dst, **const** **char** \*src);

**[cpp]** [view plain](https://blog.csdn.net/okawari_richi/article/details/57411796) [copy](https://blog.csdn.net/okawari_richi/article/details/57411796)

1. **char** \* strcpy(**char** \*dst,**const** **char** \*src)
2. {
3. **if**((dst==NULL)||(src==NULL))
5. **return** NULL;
7. **char** \*ret = dst; //[1]
9. **while** ((\*dst++=\*src++)!='\0'); //[2]
11. **return** ret;//[3]
12. }

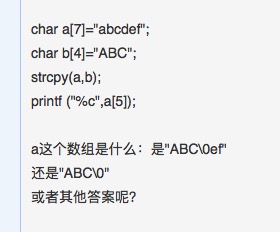
（1）const 修饰：源字符串参数用const修饰，防止修改源字符串；

（2）空指针检查：源指针和目的指针都有可能会出现空指针的情况，所以应该对其进行检查；

（3）为什么要设置ret 指针以及返回ret指针的位置[3]，由于目的指针dst已经在进行移动了，所以用辅助指针ret表明首指针；

（4）以上所示[2]处，为简单的字符串的复制过程，正好表明strcpy函数遇到'\0'将会停止；

## · 示例说明



之前在某处看到过这个问题，以下将对其进行详解，

· 最终答案输出是 "ABC\0"；

· 为什么答案不是"ABC\0ef"；

[1]从strcpy函数的实现代码可以看出当src指针指向为‘\0’时将会停止字符串的复制，由此可以得知返回ret指针所指向的数组a内容应该是 “ABC\0ef”,也就是说实际内存数组a中的内容应该是“ABC\0ef“；但是为什么最终显示会是"ABC\0"呢，原因在于，strcpy的本身属性：即strcpy只用于字符串复制，并且它不仅复制字符串内容之外，还会复制字符串的结束符；

基于此种原因，但是字符串的特性是什么呢？字符串最后一个字节存放的是一个空字符——“\0”，用来表示字符串的结束。把b复制到a之后会令b的空字符把复制后的字符串隔断,所以最终printf输出只能是"ABC\0"；

【注】此函数的第一个属性，会返回ret,也就是第一次，dst赋给ret的首地址，如

**[cpp]** [view plain](https://blog.csdn.net/okawari_richi/article/details/57411796) [copy](https://blog.csdn.net/okawari_richi/article/details/57411796)

1. **char** \*a="coda";
2. **char** b[MAX]="you are the best one.";
3. **char** \*p;
4. p=strcpy(b+8,a);
5. puts(p);

输出结果为：coda

可能到这里你已经发现了一些问题，如果想把一个字符串的一部分复制到另一个字符串的某个位置，该怎么办呢，显然strcpy()函数是满足不了这个功能的，strncpy()函数是为了弥补strcpy()函数不能检查目标字符串是否容纳下源字符串的不足而设定的一个函数。并且完全可以实现这个功能。

以上如果有错误，还请提出。

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/okawari\_richi/article/details/57411796