[C语言 realloc为什么要有返回值，realloc返回值详解/(解决任意长度字符串输入问题)。](http://blog.csdn.net/a2806005024/article/details/37656321)

标签： [realloc](http://www.csdn.net/tag/realloc)[内存](http://www.csdn.net/tag/%e5%86%85%e5%ad%98)[任意长度字符串输入](http://www.csdn.net/tag/%e4%bb%bb%e6%84%8f%e9%95%bf%e5%ba%a6%e5%ad%97%e7%ac%a6%e4%b8%b2%e8%be%93%e5%85%a5)

2014-07-10 16:13 1194人阅读 [评论](http://blog.csdn.net/a2806005024/article/details/37656321#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/a2806005024/article/details/37656321#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

Ｃ语言（3） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，未经博主允许不得转载。

    在C语言操作中会用到大量的内存操作，其中很常用的一个是realloc().

    由字面意思可以知道，该函数的作用是用于重新分配内存。

    使用方式如下：

    NewPtr=(数据类型\*)realloc(OldPtr,MemSize)

    其中OldPtr指向 待重新分配内存的指针。

    NewPtr指向 新分配空间的指针。

    MemSize为 分配后的空间大小。

    该函数的使用涉及以下几个问题：

        1.不同情况下的返回值

        2.OldPtr指向的内存会不会自动释放

        3.OldPtr和NewPtr分别是什么内容，他们有什么关系

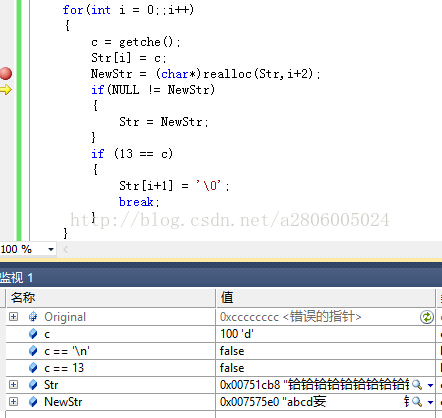
    要解答这个问题，可以从函数的动作流程出发来看：

      该函数根据内存是否满足函数需求，完成以下3个动作之一：

        1.检查OldPtr指向的内存块后面的可用内存是否满足本次realloc的需求，如果满足，则将尾部的可用内存也分配给该内存块——指针指向位置不变，内存块内容不变，指向内存块扩大。该情况下，返回值NewPtr = OldPtr;

        2.如果OldPtr指向内存块后面的可用内存不够realloc的需求。则完成以下三个步骤<a>将在可用内存中重新分配一段MemSize大小的内存，并将这段内存的开始地址返回给NewPtr。<b>将OldPtr内存块的内容复制到NewPtr内存块。<c>OldPtr的值不变，但其指向的内存块被释放（注意：是自动释放，不用主动调用free）。

笔者某次执行动作2之后的情况如图：

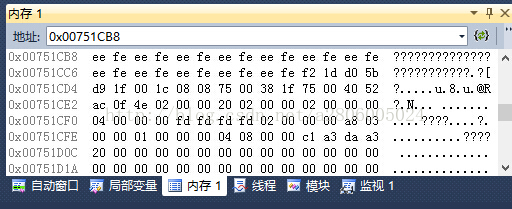


**图中程序的作用是：任意长度字符串的输入**——每轮循环输入一个字符，将该字符追加于之前输入的字符之后，使其构成一个字符串，并相应增大该字符串所占据的内存空间。

        本人依次输入了abcd....

        在输入字符abc的时候，当轮循环执行realloc之后的结果是：**1.Str和NewStr的值相同，为0x00751cb8。2.其指向的内容一致，为“abc”**

        在输入到字符d的那一轮循环，执行realloc之后的结果是，**1Str的值和NewStr的值不同了。2同时可以发现Str指向的内容被重置为“铪铪铪铪铪铪……”**查看其对应内存单元可以发现，将被free重置后的内存 0xee 0xfe 0xee 0xfe……的序列，如下图所示。

****

        3.如果整个堆当中都没有任何一处的内存能够满足realloc的需求。则此次分配失败，返回值为NULL。

        结论：在程序中realloc使用中，将OldPtr和NewPtr使用同一个指针是不妥的。使用截图对应程序中的做法较为妥当。既然能够避免分配失败返回NULL之后丢失原有指针。也能同时得到旧指针和新指针指向的内存块。