

c 中char*和char[]的区别计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022_c___E4_B8_ADchar_c97_644051.htm 问题引入：在实习过程中发现了一个以前一直默认的错误，同样char *c = "abc"和char c[]="abc"，前者改变其内容程序是会崩溃的，而后者完全正确。程序演示：
测试环境DevC 代码 #include <string.h> using namespace std. main() { char *c1 = "abc". char c2[] = "abc". char *c3 = (char*)malloc(3). c3 = "abc". printf("%d %d %s\n",amp.c2,c2,c2). printf("%d %d %s\n",&.c3,c3,c3). getchar(). } 运行结果 2293628 4199056 abc 2293624 2293624 abc 2293620 4199056 abc 参考资料：首先要搞清楚编译程序占用的内存的分区形式：一、预备知识程序的内存分配 一个由c/C 编译的程序占用的内存分为以下几个部分 1、栈区（stack）由编译器自动分配释放，存放函数的参数值，局部变量的值等。其操作方式类似于数据结构中的栈。 2、堆区（heap）一般由程序员分配释放，若程序员不释放，程序结束时可能由OS回收。注意它与数据结构中的堆是两回事，分配方式倒是类似于链表，呵呵。 3、全局区（静态区）（static）全局变量和静态变量的存储是放在一块的，初始化的全局变量和静态变量在一块区域，未初始化的全局变量和未初始化的静态变量在相邻的另一块区域。程序结束后由系统释放。 4、文字常量区常量字符串就是放在这里的。程序结束后由系统释放。 5、程序代码区 这是一个前辈写的，非常详细 //main.cpp int a=0. //全局初始化区 char *p1. //全局未初始化区 main() { int b.栈 char s[]="abc". //栈 char *p2. //栈 char *p3="123456". //123456\0在常量区， p3在栈上。 static int c=0 ;

//全局（静态）初始化区 p1 = (char*)malloc(10). p2 = (char*)malloc(20). //分配得来10和20字节的区域就在堆区。 strcpy(p1,"123456"). //123456\0放在常量区，编译器可能会将它与p3所存"123456"优化成一个地方。 } 二、堆和栈的理论知识

2.1 申请方式 stack：由系统自动分配。例如，声明在函数中一个局部变量int b；系统自动在栈中为b开辟空间 heap：需要程序员自己申请，并指明大小，在c中malloc函数如p1=(char*) malloc(10)；在C中用new运算符如p2=(char*) malloc(10)；但是注意p1、p2本身是在栈中的。

2.2 申请后系统的响应 栈：只要栈的剩余空间大于所申请空间，系统将为用户提供内存，否则将报异常提示栈溢出。 堆：首先应该知道操作系统有一个记录空闲内存地址的链表，当系统收到程序的申请时，会遍历该链表，寻找第一个空间大于所申请空间的堆结点，然后将该结点从空闲结点链表中删除，并将该结点的空间分配给程序，另外，对于大多数系统，会在这块内存空间中的首地址处记录本次分配的大小，这样，代码中的0delete语句才能正确的释放本内存空间。另外，由于找到的堆结点的大小不一定正好等于申请的大小，系统会自动的将多余的那部分重新放入空闲链表中。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com