

C编译器怎么样对内存划分和使用 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/471/2021_2022_C_E7_BC_96

[_E8_AF_91_E5_99_A8_c67_471200.htm](https://www.100test.com/kao_ti2020/471/2021_2022_C_E7_BC_96) 第一、C的编译器对内存空间的划分 这里，编译器将内存划分成4个区，分别是栈区、堆区、全局/静态区、常量区。如果要更细地划分，还可以添加一个自由存储区；把全局区看作两部分，为初始化全局区和未初始化全局区。 第二、程序中出现的不同量在内存中的保存 `int a = 0. //a：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"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。`

`//123456放在常量区，编译器可能会将它与p3所指向的"123456"优化成一块。 strcpy(p1, "123456"). }` 第三、不同内存区中的空间使用有什么特点 上述程序，已经给出了程序中出现的量，如何划分存储空间。说明如下： 1、栈 - 由编译器自动分配释放，如我们定义了一个变量，由编译器负责为它分配空间和回收。 2、堆 - 一般由程序员通过函数

（`malloc/free`等）分配释放，若程序员不释放，程序结束时可能由OS回收 3、全局区（静态区）- 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。同样由编译器分配释放，不过释放要等到程序结束时 4、常量存储区 - 编译器负责管理，程序结束释放 也就是说，C语言中，我们实际使用到，能对值发生改变的有堆、

栈和全局区，而真正能由我们自由分配和释放的操作在堆区完成。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