C中内存分区：

1. 栈：
   * 1. 由编译器自动分配释放
     2. 局部变量，函数参数等，函数结束时自动释放
     3. 栈内存分配运算内置于处理器指令中，效率高，容量有限
2. 堆：
   * 1. 有程序分配释放 new 或者 malloc
3. 全局/静态区：
   * 1. 全局变量和静态变量的存储放在一块
     2. 初始化的全局变量和静态变量放在一个区域
     3. 未初始化的全局变量和未初始化的静态变量在相邻的区域
     4. 程序结束后释放
     5. C中全局变量，静态变量 分为未初始化和初始化的，C++中没有区分
4. 常量区
   * 1. 全局常量 const修饰的全局变量
     2. 字符串常量
5. 代码区：
   * 1. 存放二进制代码

|  |
| --- |
| int a = 0; //全局初始化区 char \*p1; //全局未初始化区 void main()  {  int b; //栈  char s[] = "abc"; //栈  char \*p2; //栈  char \*p3 = "123456"; //123456{post.content}在常量区，p3在栈上  static int c = 0; //全局(静态)初始化区  p1 = (char \*)malloc(10); //分配得来得10字节的区域在堆区  p2 = (char \*)malloc(20); //分配得来得20字节的区域在堆区  strcpy(p1, "123456"); //123456{post.content}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  } |

|  |
| --- |
| IMG_256 |

从 低地址 到 高地址：

1. 预留内存区 操作系统维护的内存区域，不可访问
2. 程序代码区
3. data段 存储初始化的全局变量和static变量，还有文字常量区
4. bss段 存储未初始化的全局变量和static变量
5. 堆 由低地址 到 高地址增长 一般由new 和 malloc分配
6. 共享库文件 调用的库文件，位于堆和栈之间
7. 栈 由高地址 向 低地址增长，和堆的增长方式相对，对不同的OS来说，栈的初始化大小有规定，可以修改，一般默认是2M，由编译器自动分配释放

再上面是操作系统和内核调用的内存地址