

C 辅导:内存分配和释放 PDF转换可能丢失图片或格式，建议  
阅读原文

[https://www.100test.com/kao\\_ti2020/270/2021\\_2022\\_C\\_\\_\\_E8\\_BE\\_85\\_E5\\_AF\\_BC\\_\\_c67\\_270963.htm](https://www.100test.com/kao_ti2020/270/2021_2022_C___E8_BE_85_E5_AF_BC__c67_270963.htm) 内存分配和释放几乎是所有程序的基本要求，同时也是最容易出现问题的地方之一。通过遵循几条简单的规则，你可以避免很多常见的内存分配问题。

原则1 用new、delete取代malloc、calloc、realloc和free  
malloc、calloc、realloc和free是C语言的使用法，它们不会理会对象的存在与否，更不会去调用构造和析构函数，所以在C中经常会引起麻烦。如果混用这些函数会造成更大的麻烦。比如要防止用malloc分配，用delete释放，这些都需要花费格外的精力。

原则2 new、delete和new[]、delete[]要成对使用  
调用new所包含的动作：从系统堆中申请恰当的一块内存。若是对象，调用相应类的构造函数，并以刚申请的内存地址作为this参数。调用new[n]所包含的动作：从系统堆中申请可容纳n个对象外加一个整型的一块内存；将n记录在额外的哪个整型内存中。（其位置依赖于不同的实现，有的在申请的内存块开头，有的在末尾）；调用n次构造函数初始化这块内存中的n个连续对象。调用delete所包含的动作：若是对象，调用相应类的析构函数（在delete参数所指的内存处）；该内存返回系统堆。调用delete[]所包含的动作：从new[]记录n的地方将n值找出；调用n次析构函数析构这块内存中的n个连续对象；将这一整块内存（包括记录n的整数）归还系统堆。可以看出，分配和释放单个元素与数组的操作不同，这就是为什么一定要成对使用这些操作符的原因。

原则3 确保所有new出来的东西适时被delete掉 不需要的内存

不能及时被释放（回系统）就是大家常听到的内存泄露（memory leak）。狭义的内存泄露是指分配的内存不再使用，从更广义上说，没有及时释放也是内存泄露，只是程度较轻而已。内存泄露不管有多少，只要运行的时间够长，最终都回耗尽所有的内存。内存泄露的问题极难查找，因为当出现问题时，内存已然耗尽，此时CPU正在运行什么程序，什么程序就崩溃（哪怕是系统程序）。可以看出，崩溃时报告的错误信息与引起问题的代码毫无关系。另外内存耗尽的时间也是不确定的，且一般是一个较长的时间（几天或几个星期），这就更增加了再现和定位问题的难度。

**原则4** 自定义类的new/delete操作符一定要符合原操作符号的行为规范。比如，带同样的参数，new要返回一个内存指针，若没有足够的内存供分配则返回NULL，等等。另外，自定义类的new操作符一定要自定义类的delete操作符。

```
class Aboutmemory {
public: void* operate new(size). void operate delete(void*). void*
operate new[](size). void operate delete[](void*). }
```

**原则5** 当所有的内存被释放后，指针应该有一个合理的值，除非该指针本身要消失，否则应置为NULL。另外，千万不要忘了给字符串结束符“\0”申请一个空间！PS：所有涉及资源管理的代码应使用诸如自动指针（auto pointer）或引用计数之类的技术。

- 1.自动指针在其构造函数中接受或自己申请一个系统堆内存，并在析构函数中释放该内存。
- 2.这样，当该自动指针所在的对象消失或所在的函数退出时，自动指针的析构函数会被系统自动调用（局部变量要被系统自动释放），从而最终自动释放其所管辖的系统堆内存。实际上，这是把一个系统堆内存的生命周期限制在一贯生命周期有限的局部变量上。

3.如果异常发生，C 的异常机制保证所有已构造成功的局部变量会被自动析构。4.自动指针可以通过拷贝或赋值传递其拥有的内存：它们把所管辖的系统堆内存从源自动指针转移到目的自动指针，此后源自动指针不在拥有该系统堆内存的管辖权（析构时释放的将是一个空指针）。通过自动指针的拷贝和赋值，可以使系统堆内存能生存于申请者之外而不需要申请者释放之。5.可以看出，编程者不需要关心何时，由谁来释放申请的内存，他只需要指定下一步谁（函数，对象）要拥有该内存，并做简单的转移（拷贝或赋值）即可。拥有者内部的自动指针将完成适时的回收动作。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](http://www.100test.com)