

结合内存分布图分析内存问题计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/645/2021\\_2022\\_\\_E7\\_BB\\_93\\_E5\\_90\\_88\\_E5\\_86\\_85\\_E5\\_c97\\_645673.htm](https://www.100test.com/kao_ti2020/645/2021_2022__E7_BB_93_E5_90_88_E5_86_85_E5_c97_645673.htm)

1. 内存问题的原因及分类 在C/C++程序中，有关内存使用的问题是最难发现和解决的。这些问题可能导致程序莫名其妙地停止、崩溃，或者不断消耗内存直至资源耗尽。由于C/C++语言本身的特质和历史原因，程序员使用内存需要注意的事项较多，而且语言本身也不提供类似Java的垃圾清理机制。编程人员使用一定的工具来查找和调试内存相关问题是十分必要的。总的说来，与内存有关的问题可以分成两类：内存访问错误和内存使用错误。内存访问错误包括错误地读取内存和错误地写内存。错误地读取内存可能让你的模块返回意想不到的结果，从而导致后续的模块运行异常。错误地写内存可能导致系统崩溃。内存使用方面的错误主要是指申请的内存没有正确释放，从而使程序运行逐渐减慢，直至停止。这方面的错误由于表现比较慢很难被人工察觉。程序也许运行了很久才会耗净资源，发生问题。

### 1.1 内存解剖

一个典型的C++内存布局如下图所示：

自底向上，内存中依次存放着只读的程序代码和数据，全局变量和静态变量，堆中的动态申请变量和堆栈中的自动变量。自动变量就是在函数内声明的局部变量。当函数被调用时，它们被压入栈；当函数返回时，它们就要被弹出堆栈。堆栈的使用基本上由系统控制，用户一般不会直接对其进行控制，所以堆栈的使用还是相对安全的。动态内存是一柄双刃剑：它可以提供程序员更灵活的内存使用方法，而且有些算法没有动态内存会很难实现；但是动态内存往往是内存问

题存在的沃土。 1.2 内存访问错误 相对用户使用的语言，动态内存的申请一般由malloc/new来完成，释放由free/delete完成。基本的原则可以总结为：一对一，不混用。也就是说一个malloc必须对应一旦唯一的free；new对应一旦唯一的delete。malloc不能和delete, new不能和free对应。另外在C中要注意delete和delete[]的区别。delete用来释放单元变量，delete[]用来释放数组等集聚变量。有关这方面的详细信息可以参考[C Adv]。来源：考试大的美女编辑们 我们可以将内存访问错误大致分成以下几类：数组越界读或写、访问未初始化内存、访问已经释放的内存和重复释放内存或释放非法内存。下面的代码集中显示了上述问题的典型例子：百考试题论坛 1 #include 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)