

高质量C/C++编程指南--第7章内存管理(2) PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/134/2021\\_2022\\_\\_E9\\_AB\\_98\\_E8\\_B4\\_A8\\_E9\\_87\\_8FC\\_c97\\_134214.htm](https://www.100test.com/kao_ti2020/134/2021_2022__E9_AB_98_E8_B4_A8_E9_87_8FC_c97_134214.htm) 7.7 杜绝“野指针”

“野指针”不是NULL指针，是指向“垃圾”内存的指针。人们一般不会错用NULL指针，因为用if语句很容易判断。但是“野指针”是很危险的，if语句对它不起作用。“野指针”的成因主要有两种：（1）指针变量没有被初始化。任何指针变量刚被创建时不会自动成为NULL指针，它的缺省值是随机的，它会乱指一气。所以，指针变量在创建的同时应当被初始化，要么将指针设置为NULL，要么让它指向合法的内存。例如 `char *p = NULL. char *str = (char *) malloc(100).`（2）指针p被free或者delete之后，没有置为NULL，让人误以为p是个合法的指针。参见7.5节。（3）指针操作超越了变量的作用范围。这种情况让人防不胜防，示例程序如下：`class A { public: void Func(void){ cout }. void Test(void) { A *p. { A a. p = &a. // 注意 a 的生命期 } p->Func(). // p是“野指针” } 函数Test在执行语句p->Func()时，对象a已经消失，而p是指向a的，所以p就成了“野指针”。但奇怪的是我运行这个程序时居然没有出错，这可能与编译器有关。 7.8 有了malloc/free为什么还要new/delete？ malloc与free是C/C++语言的标准库函数，new/delete是C++的运算符。它们都可用于申请动态内存和释放内存。对于非内部数据类型的对象而言，光用malloc/free无法满足动态对象的要求。对象在创建的同时要自动执行构造函数，对象在消亡之前要自动执行析构函数。由于malloc/free是库函数而不是运算符，不在编译器控制权限之内，不能够`

把执行构造函数和析构函数的任务强加于malloc/free。因此C语言需要一个能完成动态内存分配和初始化工作的运算符new，以及一个能完成清理与释放内存工作的运算符delete。注意new/delete不是库函数。我们先看一看malloc/free和new/delete如何实现对象的动态内存管理，见示例7-8。

```
class Obj { public : Obj(void){ cout ~Obj(void){ cout void  
Initialize(void){ cout void Destroy(void){ cout }. void  
UseMallocFree(void) { Obj *a = (obj *)malloc(sizeof(obj)). // 申请  
动态内存 a->Initialize(). // 初始化 //... a->Destroy(). // 清除工作  
free(a). // 释放内存 } void UseNewDelete(void) { Obj *a = new  
Obj. // 申请动态内存并且初始化 //... delete a. // 清除并且释放  
内存 }
```

示例7-8用malloc/free和new/delete如何实现对象的动态内存管理 类Obj的函数Initialize模拟了构造函数的功能，函数Destroy模拟了析构函数的功能。函数UseMallocFree中，由于malloc/free不能执行构造函数与析构函数，必须调用成员函数Initialize和Destroy来完成初始化与清除工作。函数UseNewDelete则简单得多。所以我们不要企图用malloc/free来完成动态对象的内存管理，应该用new/delete。由于内部数据类型的“对象”没有构造与析构的过程，对它们而言malloc/free和new/delete是等价的。既然new/delete的功能完全覆盖了malloc/free，为什么C不把malloc/free淘汰出局呢？这是因为C程序经常要调用C函数，而C程序只能用malloc/free管理动态内存。如果用free释放“new创建的动态对象”，那么该对象因无法执行析构函数而可能导致程序出错。如果用delete释放“malloc申请的动态内存”，理论上讲程序不会出错，但是该程序的可读性很差。所以new/delete

必须配对使用，malloc/free也一样。7.9 内存耗尽怎么办？如果在申请动态内存时找不到足够大的内存块，malloc和new将返回NULL指针，宣告内存申请失败。通常有三种方式处理“内存耗尽”问题。（1）判断指针是否为NULL，如果是则马上用return语句终止本函数。例如：`void Func(void) { A *a = new A. if(a == NULL) { return. } ... }`（2）判断指针是否为NULL，如果是则马上用exit(1)终止整个程序的运行。例如：`void Func(void) { A *a = new A. if(a == NULL) { cout exit(1). } ... }`（3）为new和malloc设置异常处理函数。例如Visual C可以用\_set\_new\_hander函数为new设置用户自己定义的异常处理函数，也可以让malloc享用与new相同的异常处理函数。详细内容请参考C使用手册。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)