

计算机二级C_C 误区五:检查new的返回值计算机二级考试

PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/584/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E4_c97_584445.htm

2009年下半年全国计算机等级考试你准备好了没?考计算机等级考试的朋友,2009年下半年全国计算机等级考试时间是2009年9月19日至23日。

更多优质资料尽在百考试题论坛 百考试题在线题库 首先澄清一下，这个误区仅对 C 成立，这里不过是沿用“C/C 误区”

这个街头罢了。我们都知道，使用 malloc/calloc 等分配内存的函数时，一定要检查其返回值是否为“空指针”（亦即检查分配内存的操作是否成功），这是良好的编程习惯，也是编写可靠程序所必需的。但是，如果你简单地把这一招应用到 new 上，那可就不一定正确了。我经常看到类似这样的代码：

```
int* p = new int[SIZE]. if ( p == 0 ) // 检查 p 是否空指针
return -1. // 其它代码
```

其实，这里的 if (p == 0) 完全是没啥意义的。C 里，如果 new 分配内存失败，默认是抛出异常的。所以，如果分配成功，p == 0 就绝对不会成立；而如果分配失败了，也不会执行 if (p == 0)，因为分配失败时，new 就会抛出异常跳过后面的代码。如果你想检查 new 是否成功

```
，应该捕捉异常： try { int* p = new int[SIZE]. // 其它代码 }
catch ( const bad_alloc& e ) { return -1. }
```

据说一些老的编译器里，new 如果分配内存失败，是不抛出异常的（大概是因为那时 C 还没加入异常机制），而是和 malloc 一样，返回空指针。不过我从来都没遇到过 new 返回空指针的情况。当然

，标准 C 亦提供了一个方法来抑制 new 抛出异常，而返回空指针：

```
int* p = new (std::nothrow) int. // 这样如果 new 失败了，
```

就不会抛出异常，而是返回空指针 `if (p == 0) // 如此这般，`
这个判断就有意义了 `return -1. // 其它代码` 特别推荐：2009
年9月全国计算机等级考试时间及科目预告 2009年上半年全国
计算机等级考试参考答案请进入计算机考试论坛 2009年全国
计算机等级考试报名信息汇总 2009年NCRE考试有新变化
2009年全国计算机等级考试大纲 2009年上半年全国计算机二
级考试试题及答案 2009年上半年全国计算机等级考试试题答
案汇总 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