

C_C内存耗尽怎么办计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/625/2021_2022_C__C_E5_86_85_E5_AD_c97_625226.htm 编辑特别推荐: 全国计算机等级

考试（等考）指定教材 全国计算机等级考试学习视频 全国计算机等级考试网上辅导招生 全国计算机等级考试时间及科目预告 百考试题教育全国计算机等级考试在线测试平台 全国计算机等级考试资料下载 全国计算机等级考试论坛 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 << "Memory Exhausted" << endl. 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 上述（1）（2）方式使用最普遍。如果一个函数内有多处需要申请动态内存，那么方式（1）就显得力不从心（释放内存很麻烦），应该用方式（2）来处理。很多人不忍心用exit(1)，问：“不编写出错处理程序，让操作系统自己解决行不行？”不行。如果发生“内存耗尽”这样的事情，一般说来应用程序已经无药可救。如果不用exit(1) 把坏程序杀死，它可能会害死操作系统。道理如同：如果不把歹徒击毙，歹徒在老死

之前会犯下更多的罪。有一个很重要的现象要告诉大家。对于32位以上的应用程序而言，无论怎样使用malloc与new，几乎不可能导致“内存耗尽”。我在Windows 98下用Visual C编写了测试程序，见示例7-9。这个程序会无休止地运行下去，根本不会终止。因为32位操作系统支持“虚存”，内存用完了，自动用硬盘空间顶替。我只听到硬盘嘎吱嘎吱地响，Windows 98已经累得对键盘、鼠标毫无反应。我可以得出这么一个结论：对于32位以上的应用程序，“内存耗尽”错误处理程序毫无用处。这下可把Unix和Windows程序员们乐坏了：反正错误处理程序不起作用，我就不写了，省了很多麻烦。我不想误导读者，必须强调：不加错误处理将导致程序的质量很差，千万不可因小失大。

```
void main(void) { float *p = NULL. while(TRUE) { p = new float[1000000]. cout << " eat memory " << endl. if(p==NULL) exit(1). } }
```

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