new(std::nothrow) 顾名思义，即不抛出异常，当new一个对象失败时，默认设置该对象为NULL，这样可以方便的通过if(p == NULL) 来判断new操作是否成功

普通的new操作，如果分配内存失败则会抛出异常，虽然后面一般也会写上if(p == NULL) 但是实际上是自欺欺人，因为如果分配成功，p肯定不为NULL；而如果分配失败，则程序会抛出异常，if语句根本执行不到。

因此，建议在c++代码中，凡是涉及到new操作，都采用new(std::nothrow)，然后if(p==NULL)的方式进行判断

std::nothrow

在内存不足时，new (std::nothrow)并不抛出异常，而是将指针置NULL。

[关于std::nothrow](http://blog.csdn.net/hertz2007/article/details/6629654)

分类： [C++](http://blog.csdn.net/hertz2007/article/category/514663)2011-07-24 15:01 992人阅读 [评论](http://blog.csdn.net/hertz2007/article/details/6629654#comments)(0) [收藏](http://blog.csdn.net/hertz2007/article/details/6629654) [举报](http://blog.csdn.net/hertz2007/article/details/6629654#report)

http://hi.baidu.com/ilbx/blog/item/30a2c9f550299d2dbd31090f.html

nothrow new与普通new

2009-03-01 16:57

标准的new头文件可以定义普通的new，同时，它也可以定义一个变体new操作符，这个操作符叫做nothrownew。普通的new：过去和现在 普通new一个异常的类型std::bad\_alloc。这个是标准适应性态。在早期C++的舞台上，这个性态和现在的非常不同；new将返回0来指出一个失败，和malloc()非常相似。  
在一定的环境下，返回一个NULL指针来表示一个失败依然是一个不错的选择。C++标准委员会意识到这个问题，所以他们决定定义一个特别的new操作符版本，这个版本返回0表示失败。  
一个nothow new语句和普通的new语句相似，除了它的变量将涉及到std::nothrow\_t。Class std::nothrow\_t在new将按照下面的方式来定义：

**[cpp]** [view plaincopy](http://blog.csdn.net/hertz2007/article/details/6629654)

1. **class** nothrow\_t // in namespace std
2. {}; //empty class
3. Operator **nothrow** **new** is declared like **this**:
4. //declarations from <new>
5. **void** \* operator **new** (**size\_t** size, **const** std::nothrow\_t &);
6. //array version
7. **void** \* operator **new**[] (**size\_t** size, **const** std::nothrow\_t &);
8. In addition, <**new**> defines a **const** global object of type nothrow\_t:
9. **extern** **const** nothrow\_t **nothrow**; //in namespace std

按照这个方式，调用nothrow new的代码将可以使用统一的变量名字。比如：

**[cpp]** [view plaincopy](http://blog.csdn.net/hertz2007/article/details/6629654)

1. #include <new>
2. #include <iostream> // for std::cerr
3. #include <cstdlib> // for std::exit()
4. Task \* ptask = **new** (std::**nothrow**) Task;
5. **if** (!ptask)
6. {
7. std::cerr<<"allocation failure!";
8. std::exit(1);
9. }
10. //... allocation succeeded; continue normally

但是，你可以注意到你创建了你自己的nothrow\_t对象来完成相同的效应：

**[cpp]** [view plaincopy](http://blog.csdn.net/hertz2007/article/details/6629654)

1. #include <new>
2. std::nothrow\_t nt;
3. Task \* ptask = **new** (nt) Task; //user-defined argument
4. **if** (!ptask)
5. //...

分配失败是非常普通的，它们通常在植入性和不支持异常的可移动的器件中发生更频繁。因此，应用程序开发者在这个环境中使用nothrow new来替代普通的new是非常安全的。

http://www.cplusplus.com/reference/std/new/nothrow/

extern const nothrow\_t nothrow;

**Nothrow constant**

This constant value is used as an argument for [operator new](http://www.cplusplus.com/operator%20new) and [operator new[]](http://www.cplusplus.com/operator%20new%5B%5D) to indicate that these functions shall not throw an exception on failure, but return a null pointer instead.  
  
**By default, when the new operator is used to allocate memory and the handling function is unable to do so, a**[**bad\_alloc**](http://www.cplusplus.com/bad_alloc) **exception is thrown. But when nothrow is used as argument for new, it returns a null pointer instead.**  
  
This constant (nothrow) is just a value of type [nothrow\_t](http://www.cplusplus.com/nothrow_t), which only purpose is to trigger an overloaded version of function [operator new](http://www.cplusplus.com/operator%20new) or [operator new[]](http://www.cplusplus.com/operator%20new%5B%5D) that takes an argument of this type.  
  
In C++, the new operator can be overloaded to take more than one parameter: The first parameter passed to the [operator new](http://www.cplusplus.com/operator%20new) function is always the size of the element type to be allocated, but more arguments can be passed to this function by enclosing them in parentheses. For example:

**[cpp]** [view plaincopy](http://blog.csdn.net/hertz2007/article/details/6629654)

1. **int** \* p = **new** (x) **int**;

is a valid expression that, at some point, calls:

**[cpp]** [view plaincopy](http://blog.csdn.net/hertz2007/article/details/6629654)

1. operator **new** (**sizeof**(**int**),x);

By default, one of the versions of [operator new](http://www.cplusplus.com/operator%20new) is overloaded to accept a parameter of type [nothrow\_t](http://www.cplusplus.com/nothrow_t) (likenothrow). The value itself is not used, but that version of [operator new](http://www.cplusplus.com/operator%20new) shall return zero in case of failure instead of throwing an exception.  
  
The same applies for operator new[] and function [operator new[]](http://www.cplusplus.com/operator%20new%5B%5D) .

**Example**

**[cpp]** [view plaincopy](http://blog.csdn.net/hertz2007/article/details/6629654)

1. // nothrow example
2. #include <iostream>
3. #include <new>
4. **using** **namespace** std;
5. **int** main () {
6. cout << "Attempting to allocate 1 MiB...";
7. **char**\* p = **new** (**nothrow**) **char** [1048576];
8. **if** (p==0) cout << "Failed!\n";
9. **else** {
10. cout << "Success!\n";
11. **delete**[] p;
12. }
13. **return** 0;
14. }