zhangxinrun的专栏

个人资料

zhangxinrun_业余erlang

访问： 1463316次
积分： 16669
等级： BLOG 7
排名： 第204名

原创： 75篇 转载： 1065篇
译文： 0篇 评论： 141条

C++中placement new操作符（经典）

分类： C++ 2010-10-14 09:39 9361人阅读 评论(7) 收藏 举报

c++ delete class 聊天 c

placement new是重载operator new的一个标准、全局的版本，它不能被自定义的版本代替（不像普通的operator new和operator delete能够被替换成用户自定义的版本）。

它的原型如下：

```
void *operator new( size_t, void *p ) throw()  { return p; }
```

首先我们区分下几个容易混淆的关键词：new、operator new、placement new

new和delete操作符我们应该都用过，它们是对堆中的内存进行申请和释放，而这两个都是不能被重载的。要实现

不同的内存分配行为，需要重载operator new，而不是new和delete。

看如下代码：

class MyClass {…};

MyClass * p=new MyClass;

这里的new实际上是执行如下3个过程：

1调用operator new分配内存；

2调用构造函数生成类对象；

3返回相应指针。

operator new就像operator+一样，是可以重载的，但是不能在全局对原型为void operator new(size_t size)这个原型进行重载，一般只能在类中进行重载。如果类中没有重载operator new，那么调用的就是全局的::operator new来完成堆的分配。同理，operator new[]、operator delete、operator delete[]也是可以重载的，一般你重载了其中一个，那么最好把其余三个都重载一遍。

placement new是operator new的一个重载版本，只是我们很少用到它。如果你想在已经分配的内存中创建一个对象，使用new是不行的。也就是说placement new允许你在一个已经分配好的内存中（栈或堆中）构造一个新的对象。原型中void*p实际上就是指向一个已经分配好的内存缓冲区的的首地址。

我们知道使用new操作符分配内存需要在堆中查找足够大的剩余空间，这个操作速度是很慢的，而且有可能出现无法分配内存的异常（空间不够）。placement new就可以解决这个问题。我们构造对象都是在一个预先准备好了的内存缓冲区中进行，不需要查找内存，内存分配的时间是常数；而且不会出现在程序运行中途出现内存不足的异

常。所以，placement new非常适合那些对时间要求比较高，长时间运行不希望被打断的应用程序。

使用方法如下：

1. 缓冲区提前分配

可以使用堆的空间，也可以使用栈的空间，所以分配方式有如下两种：

```
class MyClass {…};
char *buf=new char[N*sizeof(MyClass)+ sizeof(int) ] ; 或者char buf[N*sizeof(MyClass)+ sizeof(int) ];
```

2. 对象的构造

```
MyClass * pClass=new(buf) MyClass;
```

3. 对象的销毁

一旦这个对象使用完毕，你必须显式的调用类的析构函数进行销毁对象。但此时内存空间不会被释放，以便其他的对象的构造。

```
pClass->~MyClass();
```

4. 内存的释放

如果缓冲区在堆中，那么调用delete[] buf;进行内存的释放；如果在栈中，那么在其作用域内有效，跳出作用域，内存自动释放。

注意：

需要学习的知识 (5)
电脑组成 (19)
手机 (4)
海量存储和算法 (21)
memcache (1)
nosql (16)
负载均衡和高可靠性 (2)
Hadoop (6)
数据挖掘-机器学习-推荐 (6)
python (28)
boost (1)
IM (5)
php (0)
javascript (2)
golang (1)
nginx (6)

文章存档

2015年01月 (1)
2014年11月 (1)
2014年10月 (3)
2014年09月 (6)
2014年06月 (3)
展开

阅读排行

linux下启动和关闭网卡命 (33442)
Django模板系统(非常详

1) 在C++标准中，对于placement operator new []有如下的说明： placement operator new[] needs implementation-defined amount of additional storage to save a size of array. 所以我们必须申请比原始对象大小多出sizeof(int)个字节来存放对象的个数，或者说数组的大小。

2) 使用方法第二步中的new才是placement new，其实是没有申请内存的，只是调用了构造函数，返回一个指向已经分配好的内存的一个指针，所以对象销毁的时候不需要调用delete释放空间，但必须调用析构函数销毁对象。

显式调用构造函数和析构函数

今天跟同事聊天，他说到STL源码有用到显示调用析构函数。试一了一下。果然能行。

结果：

Constructors

Destructors //这个是显示调用的析构函数

Destructors // 这个是delete调用的析构函数

这有什么用？

有时候，在对象的生命周期结束前，想先结束这个对象的时候就会派上用场了。

(27574)
android配置引用第三方
(16568)
推荐几个开源web自动化
(15474)
中国105个城市的必吃菜
(14232)
cookie的expires属性和
(12649)
网络广告中ctr是什么意思
(11924)
TPC,TPCC,TPMC(计算
(10464)
eclipse开发erlang (9798)
C++中placement new操 (9360)

评论排行

C++中placement new操 (7)
c语言宏定义 (5)
结构体struct的自然对齐 (5)
阿里云面试总结 (4)
解决Linux和SecureCRT (4)
漫谈C语言及如何学习C (4)
android配置引用第三方 (4)
frameset和div常规布局 (3)
Cookie文件说明及IE的C (3)
AMQP介绍 (3)

推荐文章

由此想到的：

因为我知道。

new的时候，其实做了两件事，一是：调用malloc分配所需内存，二是：调用构造函数。

delete的时候，也是做了两件事，一是：调用析造函数，二是：调用free释放内存。

所以推测构造函数也是可以显式调用的。做了个实现。

```
int _tmain(int argc, _TCHAR* argv[])
{
    MyClass* pMyClass = (MyClass*)malloc(sizeof(MyClass));
     pMyClass->MyClass();
    // …
}
```

编译pMyClass->MyClass()出错：

error C2273: 'function-style cast' : illegal as right side of '->'operator

天啊，它以为MyClass是这个类型。

解决办法有两个：

第一：pMyClass->MyClass::MyClass();
第二：new(pMyClass)MyClass();

第二种用法涉及C++ placement new 的用法。

placement new的作用就是：创建对象(调用该类的构造函数)但是不分配内存，而是在已有的内存块上面创建对象。用于需要反复创建并删除的对象上，可以降低分配释放内存的性能消耗。请查阅placement new相关资料。

显示调用构造函数有什么用？

有时候，你可能由于效率考虑要用到malloc去给类对象分配内存，因为malloc是不调用构造函数的，所以这个时候会派上用场了。

另外下面也是可以的，虽然内置类型没有构造函数。

```
int* i = (int*)malloc(sizeof(int));
```

```
new (i) int();
```

感觉这些奇奇怪怪的用法最好在写代码库时，为了达到某个目时去使用，不推荐应用开发时使用。

```
#include <iostream>
using namespace std;

class MyClass
{
public:
    MyClass()
    {
        cout << "Constructors" << endl;
    }
    ~MyClass()
    {
        cout << "Destructors" << endl;
    }

};

int _tmain(int argc, _TCHAR* argv[])
{
   MyClass* pMyClass =  new MyClass;
   pMyClass->~MyClass();
```

```
    delete pMyClass;

}
```

上一篇 关于http代理中的http connect 代理

下一篇 搜索引擎原理

查看评论

6楼 xuqing-ICT 2014-06-30 20:30发表

我想请问一下，按照placement new够早的int的类型，int是没有析构函数的吧。那么如何释放掉该内存呢？？

5楼 wanglang1000 2014-01-06 11:35发表

分析的很好。。。

4楼 v2nero 2012-11-17 18:36发表

我想这是C++实现内存管理的一个关键点吧。

3楼 大雪压 2011-01-06 09:41发表

当然这也是我的疑惑,为什么单单我这么想,难道你没有吗?

2楼 大雪压 2011-01-06 09:40发表

"需要重载operator new" 这样的话我看过很多次了,感觉就是放屁啊,operator本来就是运算符重载的标志,谈何重载operator new呢? operator new 本身就是重载new的意思.

Re: zhangxaochen 2012-09-29 13:55发表

回复weiqubo：显然“operator new” 是作为一个词组说的，“new操作符”这么叫不是很正常么。
说话、写文章的时候并不把operator当做关键字说

1楼 hw_henry2008 2010-11-06 22:33发表

[e01]谢谢楼主了！昨天刚在论坛说了句现在鄙视的话：析构函数只能自动调用。

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap