

a1875566250的专栏

目录视图 摘要视图 RSS 订阅

个人资料



a1875566250

访问：44162次
积分：954
等级：BLOG > 3
排名：千里之外

原创：47篇 转载：8篇
译文：3篇 评论：23条

文章搜索

文章存档

- 2014年12月 (1)
- 2014年11月 (1)
- 2014年10月 (10)
- 2014年09月 (1)
- 2014年07月 (1)

展开

阅读排行

VB6实现Ring3下直接调 | (3253)

[博客Markdown编辑器上线啦](#) [那些年我们追过的Wrox精品红皮计算机图书](#) [PMBOK第五版精讲视频教程](#) [火星敏捷开发1001问](#)

C++11的default和delete关键字

2014-10-23 22:50 456人阅读 评论(0) 收藏 举报

C11的新特性实在是太多了，这2个关键字关注的人倒是少了很多，其中有一个原因便是编译器支持得太慢了（VS到VS2013才支持上），不过这2个关键字那真是极为有用的，下面我们来看看。

【default关键字】

首先我们有一个字符串类：

```
[cpp]
01. class CString
02. {
03.     char* _str;
04.
05. public:
06.     //构造函数
07.     CString(const char* pstr) : _str(nullptr)
08.     {
09.         UpdateString(pstr);
10.     }
11.
12.     //析构函数
13.     ~CString()
14.     {
15.         if (_str)
16.             free(_str);
17.     }
18.
19. public:
20.     void UpdateString(const char* pstr) throw()
21.     {
22.         if (pstr == nullptr)
23.             return;
24.
25.         if ( _str)
```

- 使用Windows 8新的压缩 (1614)
- 原文翻译：绕过Windows (1480)
- 在32位Windows 8或Win (1427)
- VB6调用Windows7任务 (1333)
- Windows 8 之ntdll的内核 (1317)
- 对 Windows 中未公开的 (1277)
- CoCreateInstance 跟踪 (1266)
- ARM处理器工作模式 (1209)

评论排行

- VB6实现Ring3下直接调用 (6)
- 在32位Windows 8或Win (3)
- 关于LPC，我要说的一些 (2)
- 使用Windows 8新的压缩 (2)
- 谈Windows 8 - 传统管理 (2)
- VB6调用Windows7任务 (2)
- 使用程序更改链接到PC中 (1)
- 从传统COM简析WinRT的 (1)
- 原文翻译：绕过Windows (1)
- 程序实现Win8任务管理器 (1)

推荐文章

- * 浅析总结 Java 内部类的一些使用与梳理
- * Qt for iOS, Qt 与Objective C混合编程
- * 教你写Android ImageLoader框架之基本架构
- * 三大运营商的游戏“刷金”漏洞解决方案
- * 百度地图开发（二）之添加覆盖物 + 地理编码和反地理编码

最新评论

- 使用Windows 8新的压缩API进行梦想旅程: 最近用着win10感觉很不错
- 从传统COM简析WinRT的Async梦想旅程: 你写的文章很幽默，最近因为用上了WIN10开始学习WRT，这篇文章写的很好点赞！
- 在32位Windows 8或Windows 7杨培文: 直接把Condition="(VersionNT < v6.1) OR ((VersionNT...

```
28.         _str = (char*)malloc(strlen(pstr) + 1);
29.         strcpy(_str,pstr);
30.     }
31.
32.     public:
33.         char* GetStr() const throw()
34.         {
35.             return _str;
36.         }
37.     };
```

我们可以这样使用：

```
[cpp]
01. auto str = std::make_unique<CString>("123");
02. printf(str->GetStr());
```

但是这样是不行的：

```
[cpp]
01. auto str = std::make_unique<CString>(); //失败，因为没有有一个无参构造函数
```

好，我们用default来：

```
[cpp]
01. class CString
02. {
03.     char* _str = nullptr;
04.
05.     public:
06.         CString() = default;
07.
08.     public:
09.         //构造函数
10.         CString(const char* pstr) : _str(nullptr)
11.         {
12.             UpdateString(pstr);
13.         }
14.
15.         //析构函数
16.         ~CString()
17.         {
18.             if (_str)
19.                 free(_str);
20.         }
21.
22.     public:
23.         void UpdateString(const char* pstr) throw()
24.         {
25.             if (pstr == nullptr)
26.                 return;
27.
28.             if (_str)
```

程序实现Win8任务管理器中的获
ZL11984: 请问，这个究竟有什么
用呢？？？

VB6实现Ring3下直接调用RingC
丁国华: 谢谢分享

关于在Windows 8.1预览版上OD
qq928426122: 百度上找资料居
然找到山野哥了，哈哈。。。。

在32位Windows 8或Windows 7
a1875566250: 可以的。。。请
安装最新版本的SDK。。。。我
上图不是都建立工程了么。。。。

在32位Windows 8或Windows 7
sinservice: 这样安装后，无法建
立C#（或任何托管代码）的工程。
而C++工程则无法进行编译。看来，
原来的限制貌似是...

使用程序更改链接到PC中的Win
i_Seven_: 不错，有用...

VB6调用Windows7任务栏进度条
wk78com: 请问博主，使用进度条
后，如何清空进度条呢？发现将参
数设置为0时，还有无法清空进度
条。

```
29.         free(_str);
30.
31.         _str = (char*)malloc(strlen(pstr) + 1);
32.         strcpy(_str,pstr);
33.     }
34.
35. public:
36.     char* GetStr() const throw()
37.     {
38.         return _str;
39.     }
40. };
```

于是我们可以这样使用了：

```
[cpp] C 8
01. auto str_def = std::make_unique<CString>();
02. str_def->UpdateString(“123”);
03. printf(str_def->GetStr() == nullptr ? "None":str_def->GetStr());
```

【delete关键字】

假设我们有这样一个类，这个类是用于自动申请内存，进行RAII式管理：

（避免麻烦那什么拷贝构造拷贝赋值移动构造什么的就不写了）

```
[cpp] C 8
01. template<typename T>
02. class CStackMemoryAlloctor
03. {
04.     mutable T* _ptr;
05.
06. public:
07.     explicit CStackMemoryAlloctor(size_t size) throw() : _ptr(nullptr)
08.     {
09.         _ptr = (T*)malloc(size);
10.     }
11.
12.     ~CStackMemoryAlloctor()
13.     {
14.         if (_ptr)
15.             free(_ptr);
16.     }
17.
18. public:
19.     operator T*() const throw()
20.     {
21.         T* tmp = _ptr;
22.         _ptr = nullptr;
23.         return tmp;
24.     }
25.
26. public:
27.     T* GetPtr() const throw()
```

```
28.     {
29.         return _ptr;
30.     }
31. };
```

我们这样使用这个类：

```
[cpp] C 8

01. CStackMemoryAlloctor<wchar_t> str(128);
02. wchar_t* pstr = str.GetPtr();
03. wcscpy(pstr,L"123\n");
04. wprintf(pstr);
```

但是别人也可以这样使用：

```
[cpp] C 8

01. auto p = std::make_unique<CStackMemoryAlloctor<wchar_t>>(128);
```

因为这个类依然可以进行默认new，我们不想让人家进行new怎么办，老办法就是这样：

```
[cpp] C 8

01. private:
02.     void* operator new(std::size_t)
03.     {
04.         return nullptr;
05.     }
```

把new设置为private了，就行了，但是这样如果别人尝试new，那看到的错误提示简直惨不忍睹。。。

于是C11的delete人性化多了：

```
[cpp] C 8

01. public:
02.     void* operator new(std::size_t) = delete;
```

当尝试new的时候，提示十分友好，这个方法已被删除。

这个delete可以用来删任何你不爽的东西，比如拷贝构造，赋值拷贝什么鸡巴毛的东西。

[上一篇](#) [stdafx 头中的STRICT定义说明](#)

[下一篇](#) [谷歌C++风格的一些摘抄](#)

主题推荐

[c++11](#) [编译器](#) [strlen](#) [strcpy](#) [printf](#)

猜你在找

- [Boostboost库asio详解3io_service作为work pool](#)
- [关于muduo网络库的注解](#)
- [面试训练01时间取得栈内最小元素](#)
- [Ubuntu1204 之 CUDA 编程 二 ~~~ GPU 程序加速](#)
- [DICOM医学图像处理WEB PACS初谈四PHP DICOM Class](#)
- [在DirectX 3D中使用CEGUI](#)
- [c++ 获取操作的精确时间](#)
- [OpenGL 3ds模型显示](#)
- [Windows下MySQL C编程](#)
- [ULARGE_INTEGER运算](#)

查看评论

暂无评论

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](#)