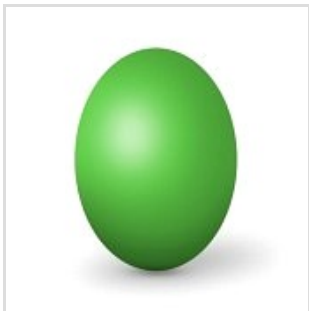


weiwenhp

追求理性之美,简单之美

个人资料



聪明的笨蛋

访问： 405981次

积分： 6753

等级： **BLOG > 6**

排名： 第1122名

原创： 271篇 转载： 3篇

译文： 1篇 评论： 71条

文章分类

- 【一】Software Test (1)
  - >(1)Test Theory (0)
  - >(2)White-box Test (0)
  - >(3)Automated Test (0)
  - >(4)Performance Test (0)
- 【二】Software Dev (1)
  - >(1) Python (2)
  - >(2) Java (1)
  - >(3)JavaScript (0)
  - >(4)PHP (0)
  - >(5) C++ (60)

[从零开始掌握iOS8开发技术（Swift版）](#) [那些年我们追过的Wrox精品红皮计算机图书](#) [CSDN学院--学习礼包大派送](#) [CSDN JOB](#)  
[带你坐飞机回家过年](#)

## C++ 内连接与外连接

分类： ----->(5) C++

2013-02-21 15:23

1212人阅读

评论(2)

收藏

举报

目录(?)

[+]

啥叫内连接 外连接

我们知道编译的时候(假如编译器是VS),是以源文件cpp文件为单位,编译成一个个的obj文件,然后再通过链接器把不同的obj文件链接起来.

简单的说,如果一些变量或函数的定义是内连接的话,链接器链接的时候就不会拿它们去与obj比较看有重复定义不,一个源文件中的extern声明的变量或函数也不能使用另外一个源文件中的内连接的变量或函数.而如果是外连接的话则需要不同的obj中比较是否有重定义的.除了做这样的检查外,链接器还会查看通过extern修饰的变量或函数声明在其他obj中的定义.

### cpp源文件与h头文件关系

我们知道C++支持多种编程范式,可以完全用采用面向过程,不去用类,当然了很少有人这样做,一般是结合面向过程与面向对象.

那假如有一些处在不同源文件中的变量和函数(不是类中的变量成员或成员函数),我们要使用不同源文件中的变量或函数时咋整呢? 在面向对象中我们使用一个个类的时候自然是用头文件引用下就OK.但现在没有类,只是一个个变量和函数,也能用引用头文件吗? 这得看情况,有时可以,有时不行.如果头文件中只有外部声明没有任何定义,那引用头文件完全没有问题.如果头文件中有定义的话,如果只被一个cpp文件引用则没问题,如果被多个cpp文件引用就会出现重复定义的错误.(注:编译器是以cpp文件为单位编译,如果某个h头文件没有被引用的话相当于被抛弃不用了.引用头文件时预编译时只是简单的把头文件复制到引用它的cpp文件中.)

----->(6) C# (57)

----->(7) MFC (52)

【三】Hardware (0)

【四】Linux (9)

【五】Database (0)

->(1)MySQL (0)

-->(2) Oracle (42)

--->(3) SQL Server (7)

---->(4) Other DB (5)

【六】Others (0)

->(1)English Study (28)

-->(2) Computer Theory (20)

--->(3) Data Structure (21)

---->(4)IT Related (0)

----->(5) Essay (11)

文章搜索

文章存档

2014年12月 (2)

2014年09月 (2)

2014年07月 (4)

2014年03月 (1)

2014年02月 (3)

展开

最新评论

什么叫编译时和运行时

congleetea: 不错，正在纳闷动态联编和静态联编的问题。这个很有帮助！

智能指针(二):shared\_ptr实现原理

sky\_ting: @lr\_zhaohj:对 我也正好设计一个shared\_ptr模板类，用指针确实比较妙。

我的新工作感悟

阿呆阿瓜: 同感，时间不够用，也在想紧急重要这些东西，可还是感觉一团糟。。

什么叫编译时和运行时

纺大才子: 那些踩的人是什么心态？这不写得挺生动，挺简单易懂的么？要是觉得哪里不好，就该在评论里指出来，而不是踩...

什么叫编译时和运行时

u010615675: 学习了，学了好长时间C语言了，没认真想过这个问题！

智能指针(二):shared\_ptr实现原理

zhshuai1: 我也觉得引用计数应该使用指针指向同一个对象

## extern外部声明

假如在有one.cpp和two.cpp两个源文件.

```
//one.cpp中

-----

#include <iostream>

using namespace std;

int number = 123; //number的定义 .或者写成extern number = 123;当有赋值时,实际上extern失去了应有的作用.
所以加不加没影响.

void Print() {cout<<"hi,i am one.cpp"<<endl;}

-----

//two.cpp中

-----

#include <iostream>

using namespace std;

extern int number;//这就是所谓的外部声明,此处extern不可省.另外此处绝对不能赋值.如果写成extern int number
= 88;会报错是重复定义.

extern void Print(); //此处extern可以省略.

cout<<number; //结果为123

Print(); //输出i am one.cpp
```

**extern int number;**//这就是所谓的外部声明,此处extern不可省.另外此处绝对不能赋值.如果写成extern int number = 88;会报错是重复定义.

**extern void Print();** //此处extern可以省略.

cout<<number; //结果为123

Print(); //输出i am one.cpp

在two.cpp中是怎么得到one.cpp中的number的值的呢,由于用extern int number这样声明了下,表明number在其他源文件中有定义,链接器就会帮助去其他源文件中找的.

假如把上面two.cpp中的extern关键字去掉. 编译时不会出错.但链接时出错了,重复定义了.因为one.cpp中已经定义了一个number,不能再定义一个相同的了.

MFC序列化(serial)简单例子  
icelolipop: 不错额

MFC序列化(serial)简单例子  
icelolipop: 这个代码。。。

MFC,QT与WinForm,WPF简介  
devil\_zhuzhu: 赞！

MFC怎么封装CreateWindow  
hhsguo: 和我遇到了一样的问题，我找了半天终于找到你这里了，可是当我看到最后一句的时候我失望了，默认回调函数是...

## static 内部连接

上面的例子中我们知道one.cpp和two.cpp中同时写上int number会出错,说重复定义了.但如果这样

//one.cpp中

static int number = 123;

//two.cpp中

static int number; //没显式赋值,会默认赋予0

此时却不会出错.因为定义变量时默认是外部连接的.而加上关键字static表示是静态变量,是内部连接,链接器不会去看不同cpp编译成的obj文件中有重名的静态变量不.

当用static修饰后就不能再使用extern修饰了.

//one.cpp中

static int number = 123;

//two.cpp中

extern int number;

cout<<number;

此时会出错,因为extern声明的number找不到定义.因为one.cpp的number用static修饰表明是内连接了.

## const关键字

//one.cpp中

const int number = 123;

//two.cpp中

const int number = 321;

这里达到的效果与static一样,都属于内部连接,所以不会出错.唯一不同的时const表示常量,定义时必须显式赋予值,

且赋值后不能再改变它的值.

不过**const**还有另外一个特性就是可以和**extern**一起用.

比如在two.cpp中这样写

```
extern const int number;
```

```
cout<<number; //运行会报错,会说找不到定义.需要把one.cpp中改为extern const int number = 123;才行.
```

```
//正确输出的值是one.cpp中的number值123
```

### **inline**与**static**函数也是内部连接的

```
//one.cpp中
```

```
void Test() { }
```

```
//two.cpp中
```

```
void Test() { }
```

这样编译时会报错,重复定义了.但如果把上面的两个**void Test**都改成**inline void Test() { }**或者**static void Test() { }**则不会出错.//注意这里讲的**inline**函数指的是全局函数,不是类里面的**inline**函数.

所以函数跟一般变量差不多.没任何修饰的就默认是外部连接,有**static**修饰的则是内部连接.另外没有**const**函数这一说,只有在类中才可以在函数后面加个**const**来修饰

[上一篇](#) [日语五十音图详解](#)

[下一篇](#) [排序算法简介](#)

顶

1

踩

0

主题推荐

[c++](#)

[面向对象](#)

[namespace](#)

[编译器](#)

[iostream](#)

猜你在找

实习生招聘笔试	Leetcode全链表问题
C++实现单例模式的两种方法	通过数组指针取数组元素的值
Qt中右击菜单treeWidget	用数组实现链表一静态链表
长程调度中程调度短程调度	c++ 类中static变量初始化问题
顺序栈的实现	许小乖的IT之路2014一

准备好了么？跳吧！更多职位尽在 CSDN JOB

软件开发工程师（C++）	我要跳槽	C++软件开发工程师	我要跳槽
苏州敏行医学信息技术有限公司	6-10K/月	天津市努思企业服务有限公司	7-10K/月
软件工程师（C++/C#）	我要跳槽	C++开发工程师	我要跳槽
武汉海翼科技有限公司	4-6K/月	搜狐畅游	10-20K/月

查看评论

2楼 Elvin\_C\_L 2014-03-31 09:42发表



不错，学习了

1楼 虾米ning 2014-01-16 19:41发表



涨姿势

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

