

Script

脚本之家

www.jb51.net

单号网

空包代发

免费试用，满意购买

自定义签收人，淘宝物流显示，下单后底单五分钟可自助下载

51IDC 香港国际数据中心

服务器租用 服务器托管

自有AS号|大陆专线|

MPLS VPN 全新上线

首页

网页制作

脚本专栏

脚本下载

网络编程

数据库

CMS教程

电子书籍

平面设计

媒体动画

操作系统

网站运营

网络安全

C#教程

vb

vb.net

C 语言

Java编程

Delphi

java

Android

IOS

Swift

Scala

其它相关

特色栏目：

vbscript

正则表达式

javascript

批处理

服务器软件

素材下载

搜索

JD 京东

11.11 天



¥ 249.00

3657好评

苏泊尔supor砂锅·石锅·陶瓷煲·新陶养生煲4.5L·



¥ 69.90

4094好评 赠京豆

滇虹康王清养去屑洗发
水套装400毫升+200毫



¥ 79.00

9354好评

小熊 (Bear) DRG-210GA 电煮锅 不锈钢

低价出售流量10000IP只需8元

装软件赚钱，每台电脑秒赚6.3元

众生网络 基于云计算的互联网基础服务运

枫信科技-江苏双线10M保证-399/元

鼎点网络百兆独享服务器仅需999元

【阿里云SSD云盘】速度行业领先

VeryCloud 专业cdn加速服务商

服务器租用/托管-域名空间/认准腾佑科技

四核15M独享服务器特价188元/月

免备案国内虚拟主机-163cdn

网站加速(加速乐)内部1折券【数量有限】

微子网络 湛江双线高防 抗15G 700元起

BGP四线 亿恩1U服务器托管3999元/年

免备vps20/百独799/双线350/45互联

畅游网络 百独服务器 包跑满 998元

香港VPS|美国VPS 仅售28元 买1年送1年

免费VPN秒进谷歌，3KVPN秒看youtube

九九数据 — 工信部认可正规资质IDC接入商

VeryCloud云主机，仅需29元，抢疯了！

服务器租用 199/月起

西部数码

www.west263.com

中国云主机领导品牌

弹性云主机

故障百倍赔偿

香港云服务器

98元/月

广告位招租

免备案 香港空间

阿里云

aliyun.com

云服务器9.9元

学生专享

阿里云助你实现创业梦想

页面导航：

首页

→

软件编程

→

C 语言

→

正文内容

默认构造函数

从汇编看c++中默认构造函数的使用分析

作者：

字体：[增加 减小]

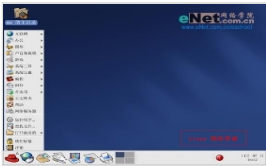
类型：转载

时间：2013-05-06

c++中，如果为一个类没有明确定义一个构造函数，那么，编译器就会自动合成一个默认的构造函数。下面，通过汇编程序，来看一下其真实情况



新款大众途观



linux入门



高能点火器



科雷傲报价

您现在的位置

个人住房公积金查询姓名

翼虎最新报价

别克昂科威

情侣酒店

软件编程软件

软件编程工具

金太阳炒股

医疗保险个人账户查询系统

西安最新楼盘

delphi

途观报价

途观越野车报价

深圳晚报就偷拍致歉

翼虎轮胎

标致408

c++中的源程序：

代码如下：

[复制代码](#)

```
class X {
private:
    int i;
};

int main() {
    X x;
}
```

上面的类X没有定义构造函数，仅仅有一个int i。

下面为其汇编程序：

代码如下：

[复制代码](#)

```
; 7 : int main() {

    push    ebp;ebp为一个寄存器，总是指向一个函数调用堆栈的栈底，作为基址，用偏移量来访问该调用栈上的变量，但这里没有任何变量要访问，因此不起作用

    mov     ebp, esp;这两句的作用是为了保存调用main之前堆栈的基址ebp的值，并将ebp指向main调用栈的栈底
```

最近更新

tcp socket客户端和服务端示例分享

深入分析C语言分解质因数的实现方法

C++ 基本算法 冒泡法、交换法、选择法、实

基于memset()函数的深入理解

MySQL的C语言API接口

wchar_t,char,string,wstring之间的相互转

探讨C语言中关键字volatile的含义

C语言时间处理实例分享

STL各个容器性能详细比较

C++设计模式之桥梁模式

阿里云

aliyun.com

一键部署

Web环境

轻松搞定云服务器配置



热点排行

- 1
- c语言10个经典小程序
- 2
- C语言字符串操作总结大全(超详细
- 3
- C语言程序设计50例(经典收藏)
- 4
- c语言字符数组与字符串的使用详解
- 5
- C语言运算符优先级列表(超详细)
- 6
- C语言文件操作函数大全(超详细)
- 7
- C语言/C++中如何产生随机数
- 8
- c语言连接mysql数据库的实现方法
- 9
- fatal error LNK1104: 无法打开文
- 10
- string,CString,char*之间的转化

```
push    ecx;将寄存器ecx的值压栈, 栈顶指针esp向前移动4byte
;这句的作用, 为即将要创建的对象预留了4byte的空间, 并向里面写入ecx的值

; 8   :   X x;
; 9   : }

xor     eax, eax;eax也是一个寄存器, 这里不起作用
mov     esp, ebp;将栈顶指针移动到push ecx前的位置, 即释放了4byte的空间
pop     ebp;恢复基址到main调用之前的状态
ret     0;函数返回
```

通过汇编发现, 通过push ecx, 编译器将堆栈栈顶移动4byte, 并将寄存器的ecx的值写入, 类X只含有一个int, 大小刚好为4byte, 因此这一句可以看成是为对象x分配空间。而接下来并没有任何函数的调用, 来对这一块区域进行适当的初始化。所以, 在没有明确定义一个构造函数的时候, 不会有任何的初始化操作。

下面再看一段c++程序：

代码如下:	复制代码
<pre>class X { private: int i; int j;//增加一个成员变量int j }; int main() { X x; }</pre>	

与上面相比, 在类X里面增加了一个成员变量int j, 类的大小变为8字节。

下面为对应汇编码：

代码如下:	复制代码
<pre>; 8 : int main() { push ebp mov ebp, esp sub esp, 8; 栈顶指针移动8byte, 刚好等于类X的大小 ; 9 : X x; ; 10 : } xor eax, eax mov esp, ebp pop ebp ret 0</pre>	

从汇编码看出, 通过sub esp, 8指令, 堆栈确实留出了8byte的空间, 刚好等于类X的大小, 同样没有调用任何函数, 来进行初始化操作。

所以, 综上所述, 在一个类没有明确定义构造函数的时候, 编译器不会有任何的函数调用来进行初始化操作, 仅仅是移动栈顶留出对象所需空间, 也就是说, 这种情况下, 编译器根本不会提供默认的构造函数。

那么, 书上说的由编译器提供默认的构造函数到底是怎么回事呢？

下面看第一种情况, 类里面有虚成员函数：

c++源码如下：

代码如下:	复制代码
<pre>class X { private: int i; int j;//增加一个成员变量int j public: virtual ~X() {</pre>	



321 Networks

仅限1000台，租完为止

电信、联通、移动、单/双/三线机房任其选

销售：52666666

Js与CSS工具

CSS在线压缩格式化(中文)

css 格式化整理工具(英文)

CSS整形格式化

JavaScript 格式化整理工具

jsbeautifier Js格式化整理工具(英文)

php 格式化整理工具(英文)

HTML/JS互相转换工具

javascript pack加密压缩工具

JS Minifier压缩

JS混淆工具

在线JS脚本校验器错误

JavaScript 正则表达式在线测试工具

EN

0元

免费试用

VPS主机,购买立即赠送内存

只需99元起

集思网络

高端机房，一线资源，价格低廉

代码转换工具

Base64编码加密

Escape加解密

HTML/UBB代码转换

GB2312/BIG5繁简字转换

经典小工具集 数字转换

HTML多功能代码转换器

迅雷快车加/解密

汉字转换拼音

租用 返3000

丽水双线托管4999元

枫信科技

【百独线路促销】

活动一：江苏电信1U百独托管 999月

100%纯百独 可测试 满意后付款

```
    }  
};  
  
int main() {  
    X x;  
}
```

析构函数为虚函数

下面是**main**函数对应的汇编码：

代码如下:	复制代码
<pre>; 13 : int main() { push ebp mov ebp, esp sub esp, 12 ; 为对象x预留12byte的空间，成员变量int i, int j占8byte，由于有虚函数，因此vptr指针占4byte ; 14 : X x; lea ecx, DWORD PTR _x\$[ebp];获取x对象的首地址，存入ecx寄存器 call ??0X@@QAE@XZ;这里调用x的构造函数 ; 15 : } lea ecx, DWORD PTR _x\$[ebp];获取对象x的首地址 call ??1X@@UAE@XZ ; 调用析构函数 xor eax, eax mov esp, ebp pop ebp ret 0</pre>	

可以看到，对象x的构造函数被调用了，编译器确实合成了默认的构造函数。

下面是构造函数的汇编码：

代码如下:	复制代码
<pre>??0X@@QAE@XZ PROC ; X::X, COMDAT ; _this\$ = ecx push ebp mov ebp, esp push ecx mov DWORD PTR _this\$[ebp], ecx;ecx寄存器存有对象x的首地址 mov eax, DWORD PTR _this\$[ebp];将对象x的首地址给寄存器eax mov DWORD PTR [eax], OFFSET ??_7X@@@6B@;这里设置vptr指针的值，指向vtable (OFFSET ??_7X@@@6B@是获得vtable的地址) ;并且通过这句，也可以证明vptr指针位于对象其实地址处 mov eax, DWORD PTR _this\$[ebp] mov esp, ebp pop ebp ret 0</pre>	

可以看到，由于有虚函数，涉及到多态，因此构函数初始化了vptr指针，但是没有为另外两个变量int i, int j赋值。

从上面可以看出，类里面含有虚函数时，在没有明确定义构造函数时，编译器确实会为我们提供一个默认的构造函数。因此当一个类继承自虚基类时，也满足上面的情形。

接下来是第二种情形，类Y继承自类X，X明确定义了一个默认的构造函数(并非编译器提供),而类Y不定义任何构造函数：

先来看看**c++**源码：

代码如下:	复制代码
<pre></pre>	

```
class X {
private:
    int i;
    int j;
public:
    X() { //X显示定义的默认构造函数
        i = 0;
        j = 1;
    }
};

class Y : public X { //Y继承自X
private:
    int i;
};

int main() {
    Y y;
}
```

类Y里面没有显示定义任何构造函数

下面是**main**函数对应的汇编码：

代码如下:	复制代码
<pre>; 19 : int main() { push ebp mov ebp, esp sub esp, 12 ; 为对象y预留12byte空间，y自身成员变量int i占4byte 父类中的成员变量int i int j占8byte ; 20 : Y y; lea ecx, DWORD PTR _y\$[ebp]; 获取对象y的首地址，存入寄存器ecx call ??0Y@@QAE@XZ; 调用对象y的构造函数 ; 21 : }</pre>	

main函数中调用了由编译器提供的默认y对象的默认构造函数。

下面是编译器提供的**y**对象默认构造函数的汇编码：

代码如下:	复制代码
<pre>??0Y@@QAE@XZ PROC ; Y::Y, COMDAT ; _this\$ = ecx push ebp mov ebp, esp push ecx mov DWORD PTR _this\$[ebp], ecx; ecx中存有对象y的首地址 mov ecx, DWORD PTR _this\$[ebp] call ??0X@@QAE@XZ ; 调用父类X的构造函数 mov eax, DWORD PTR _this\$[ebp] mov esp, ebp pop ebp ret 0 ??0Y@@QAE@XZ ENDP</pre>	

可以看到y对象的构造函数又调用了父类的构造函数来初始化继承自父类的成员变量，但自身成员变量依然没有初始化。

下面是父类**X**的构造函数汇编码：

代码如下：[复制代码](#)

```
; 7 : X() {  
  
    push    ebp  
    mov     ebp, esp  
    push    ecx  
    mov     DWORD PTR _this$[ebp], ecx  
  
; 8 :     i = 0;  
  
    mov     eax, DWORD PTR _this$[ebp]  
    mov     DWORD PTR [eax], 0; 初始化成员变量i  
  
; 9 :     j = 1;  
  
    mov     ecx, DWORD PTR _this$[ebp]  
    mov     DWORD PTR [ecx+4], 1; 初始化成员变量j  
  
    mov     eax, DWORD PTR _this$[ebp]  
    mov     esp, ebp  
    pop     ebp  
    ret     0  
}
```



在父类**X**的存储空间中，从首地址开始的8比特用来存储继承自父对象的成员变量*i*，后4byte用来存储自己的成员变量*j*。

由于首地址存储了父类成员变量*i*，因此内存地址要从对象**y**的首地址要移动4byte，才能找到父类成员变量*j*所处位置。

可以看到，**y**对象继承自父类的成员变量由父类构造函数初始化。父对象包含在子对象中，并且**this**指针，即寄存器**ecx**存储的首地址始终是子对象**y**的首地址。

如果父类**X**中也没有定义任何构造函数会怎样？

下面是**C++**源码：

代码如下：[复制代码](#)

```
class X {  
private:  
    int i;  
    int j;  
  
};  
  
class Y : public X{//Y继承自X  
private:  
    int i;  
};  
  
int main() {  
    Y y;  
}
```

父类和子类都没有任何构造函数。

下面是**main**函数汇编码：

代码如下：[复制代码](#)

```
; 16 : int main() {  
  
    push    ebp  
    mov     ebp, esp  
    sub     esp, 12 ; 和刚才一样，为对象y预留12byte  
  
; 17 :     Y y;
```

```
; 18 : }

xor    eax, eax
mov    esp, ebp
pop    ebp
ret    0
```

可以看到main中根本没有任何函数的调用，也就是说，编译器没有为子对象y提供默认构造函数。

那么，要是父类中带参数的构造函数，而子类中没有构造函数呢?这时候编译器会报错。

下面看第三种情况，类Y中包含成员对象X，成员对象有显示定义的默认构造函数，而类Y没有任何构造函数：

先看c++源码：

代码如下:	复制代码
<pre>; 16 : int main() { push ebp mov ebp, esp sub esp, 12 ; 和刚才一样，为对象y预留12byte ; 17 : Y y; ; 18 : }</pre> <pre> xor eax, eax mov esp, ebp pop ebp ret 0</pre>	

类X为类Y的成员对象

下面是main函数的汇编码：

代码如下:	复制代码
<pre>; 21 : int main() { push ebp mov ebp, esp sub esp, 12 ; 为对象y预留12byte 成员对象的变量占8byte 对象y自身占变量占4byte 成员对象包含在对象y中 ; 22 : Y y; lea ecx, DWORD PTR _y\$[ebp];对象y的首地址存入ecx call ??0Y@@@QAE@XZ;调用对象y的构造函数，由编译器提供的默认构造函数 ; 23 : }</pre> <pre> xor eax, eax mov esp, ebp pop ebp ret 0</pre>	

对象y的构造函数被调用，即编译器提供了默认的构造函数

对象y的构造函数汇编码：

代码如下:	复制代码
<pre>??0Y@@@QAE@XZ PROC ; Y::Y, COMDAT ; _this\$ = ecx push ebp mov ebp, esp push ecx mov DWORD PTR _this\$[ebp], ecx;ecx中存有对象y的首地址 mov ecx, DWORD PTR _this\$[ebp] add ecx, 4;加4是因为对象y首地址起始处存储的是自身成员变量i call ??0X@@@QAE@XZ ; 调用成员对象x的构造函数</pre>	

```
mov    eax, DWORD PTR _this$[ebp]
mov    esp, ebp
pop    ebp
ret    0
```

对象y的构造函数调用了成员对象x的构造函数，用来初始化成员对象中的成员变量，对象y自身的成员变量没有初始化。

成员对象x的构造函数汇编码：

代码如下:	复制代码
<pre>??0X@@@QAE@XZ PROC ; X::X, COMDAT ; _this\$ = ecx ; 7 : X() { push ebp mov ebp, esp push ecx mov DWORD PTR _this\$[ebp], ecx;ecx中存有成员对象x的起始地址 ; 8 : i = 0; mov eax, DWORD PTR _this\$[ebp];成员对象x的起始地址给eax寄存器 mov DWORD PTR [eax], 0;初始化成员对象x中额成员变量i ; 9 : j = 0; mov ecx, DWORD PTR _this\$[ebp];成员对象x的起始地址给ecx寄存器 mov DWORD PTR [ecx+4], 0;初始化成员对象x中额成员变量j 加4的原因是j的地址偏离了成员对象x 起始地址4byte(即成员对象x的成员变量i的字节数) ; 10 : } mov eax, DWORD PTR _this\$[ebp] mov esp, ebp pop ebp ret 0</pre>	

但是，如果成员对象x没有任何构造函数，情形会怎样呢？

下面是c++源码：

代码如下:	复制代码
<pre>class X { private: int i; int j; }; class Y { private: int i; X x;//x成员对象 }; int main() { Y y; }</pre>	

下面是main函数汇编码：

代码如下:	复制代码
<pre>; 17 : int main() {</pre>	

```
push    ebp
mov     ebp, esp
sub     esp, 12           ; 为对象预留12byte空间

; 18 :   Y y;
; 19 : }

xor     eax, eax
mov     esp, ebp
pop     ebp
ret     0
```

可以看到，main函数里面没有任何函数调用，也就是说编译器没有提供默认构造函数。

那要是成员对象x有带参数的构造函数(即非默认构造函数)，而对象y没有任何构造函数呢？此时，编译器会报错。

这种情形和前一种情形很相似。

综合以上的情况，可以总结出，对于一个类不含任何构造函数，而编译器会提供默认的构造函数，有一下3种情形：

- 1 类本身函数虚成员函数或者继承自虚基类
- 2 类的基类有构造函数，并且基类构造函数还是显示定义的默认构造函数(非编译器提供),若基类的构造函数带有参数(即非默认构造函数),编译器报错
- 3 这种情况和上一种相似，类的成员对象有构造函数，并且成员对象的构造函数还是显示定义的默认构造函数(非编译器提供);若成员对象的构造函数带有参数(即非默认构造函数),编译器报错。

以上参考了《VC++深入详解》里面的知识点，还有自己的分析，欢迎指正

您可能感兴趣的文章：

- C++类成员构造函数和析构函数顺序示例详细讲解
- c++基础语法:构造函数与析构函数
- c++基础语法:构造函数初始化列表
- C++中拷贝构造函数的总结详解
- C++拷贝构造函数(深拷贝与浅拷贝)详解
- 深入解析C++中的构造函数和析构函数
- c++ 构造函数的初始化列表
- c++中拷贝构造函数的参数类型必须是引用
- 深入C++中构造函数、拷贝构造函数、赋值操作符、析构函数的调用过程总结
- 深入C++拷贝构造函数的总结详解
- c++拷贝构造函数防篡改示例



新款大众途观



游戏编程



游戏编程入门



linux入门



别克昂科威

住房公积金明细查询图解

标致508报价

2014养老金个人账户余额

东风标致508

保险公司排名

c++工程师

翼虎最新报价

编程入门教程

软件工程师

20万suv推荐

您现在的位置

个人住房公积金查询姓名

教官女生宿舍楼下杀狗烤肉

标致3008

途观越野车报价

西安最新楼盘

情侣酒店

买二手车

翼虎轮胎

别克suv报价

Tags: C++ 默认构造函数

相关文章

C语言中设置进程优先顺序的方法	2015-08-08
c++友元函数与友元类的深入解析	2013-07-07
C语言的getc()函数和gets()函数的使用对比	2015-08-08
C/C++ 宏详细解析	2013-09-09

VC++中HTControl控件类的CHTSlider控制杆控件类简介	2014-08-08
C++ vector删除符合条件的元素示例分享	2014-02-02
一些语言的按行读取文件的代码实现小结	2015-08-08
深入理解:Java是类型安全的语言,而C++是非类型安全的语言	2013-06-06
VC++植物大战僵尸中文版修改器实现代码	2015-04-04
C++之异常处理详解	2013-08-08



个人账户 领取



新款大众途观



游戏编程



离合器



东风标致



软件编程



c++语言



一汽大众途观



途观汽车



c语言入门



途观汽车价格



雷诺汽车

文章评论

友情提醒：本站文件的解压密码：www.jb51.net (请使用最新的winrar)



猜你喜欢

男人多长，多久才正常！有图

颈椎病。头晕。手麻。一招治好

颈椎病。头晕。手麻。一招治好

深入Main函数中的参数argc,argv的使用详解

通过c++11改进我们的模式之改进命令模式

构造函数定义为private或者protected的好处

C#私有构造函数使用示例

深入探讨C++父类子类中虚函数的应用

用c++实现x的y次幂的代码

c++与c#的时间转换示例分享

c#只读字段和常量的区别,以及静态构造函数的使用

c++函数中的指针参数与地址参数区别介绍

百度推荐