



方亮的专栏

目录视图

摘要视图

RSS 订阅

个人资料



breaksoftware



访问: 673867次

积分: 8696

等级: 5

排名: 第2191名

原创: 201篇

转载: 1篇

译文: 0篇

评论: 441条

文章搜索



博客专栏

IT项目研发过程中的利器
文章:4篇
阅读:442libev源码解析
文章:6篇
阅读:676PE文件和COFF文件格式分析
文章:11篇
阅读:28979DllMain中不当操作导致死锁问题的分析
文章:9篇
阅读:36501WMI技术介绍和应用
文章:24篇
阅读:110414GTest源码解析
文章:11篇
阅读:25231

文章分类

DllMain中的做与不做 (9)

赠书 | 异步2周年,技术图书免费送 每周荐书:渗透测试、K8s、架构(评论送书) 项目管理+代码托管+文档协作,开发更流畅

Google Test(GTest)使用方法和源码解析——私有属性代码测试技术分析

2016-04-07 23:58

1449人阅读

评论(0)

收藏 举报

分类: GTest使用方法和源码解析 (10)

版权声明:本文为博主原创文章,未经博主允许不得转载。

有些时候,我们不仅要测试类暴露出来的公有方法,还要测试其受保护的或者私有方法。GTest提供了一种方法,让我们可以测试类的私有方法。但是这是一种侵入式的,会破坏原来代码的结构,所以我!

用。(转载请注明出于breaksoftware的csdn博客)

我们先看个例子

```
[cpp]
01. // This class has a private member we want to test. We will test it
02. // both in a TEST and in a TEST_F.
03. class Foo {
04. public:
05.     Foo() {}
06.
07. private:
08.     int Bar() const { return 1; }
09.
10.     // Declares the friend tests that can access the private member
11.     // Bar().
12.     FRIEND_TEST(FRIEND_TEST_Test, TEST);
13.     FRIEND_TEST(FRIEND_TEST_Test2, TEST_F);
14. };
15.
16. // Tests that the FRIEND_TEST declaration allows a TEST to access a
17. // class's private members. This should compile.
18. TEST(FRIEND_TEST_Test, TEST) {
19.     ASSERT_EQ(1, Foo().Bar());
20. }
21.
22. // The fixture needed to test using FRIEND_TEST with TEST_F.
23. class FRIEND_TEST_Test2 : public Test {
24. protected:
25.     Foo foo;
26. };
27.
28. // Tests that the FRIEND_TEST declaration allows a TEST_F to access a
29. // class's private members. This should compile.
30. TEST_F(FRIEND_TEST_Test2, TEST_F) {
31.     ASSERT_EQ(1, foo.Bar());
32. }
```

Foo类拥有一个名叫Bar的私有成员函数。我们需要通过FRIEND_TEST宏来新增该类的友元类(其实也能想象出来,就是通过友元类来访问私有成员,所以这是一种侵入式的——修改了原来类的结构)

```
[cpp]
01. #define FRIEND_TEST(test_case_name, test_name)\
02.     friend class test_case_name##_##test_name##_Test
```

上例中给了两种访问私有变量的方式:

1. 一种是FRIEND_TEST(FRIEND_TEST_Test, TEST);声明的类,然后通过TEST(FRIEND_TEST_Test, TEST)实现测试特例实体,于是该实体中需要使用Foo的构造函数构造一个对象,然后调用其私有的Bar方法。
2. 一种是FRIEND_TEST(FRIEND_TEST_Test2, TEST_F);声明的类。它使用到了Test Fixtures技术(详见《Google Test(GTest)使用方法和源码解析——预处理技术分析和应用》)。在FRIEND_TEST_Test2类中,声

WMI技术介绍和应用 (24)
Apache服务搭建和插件实现 (7)
网络编程模型的分析、实现和对比 (6)
GTest使用方法和源码解析 (11)
PE文件结构和相关应用 (11)
windows安全 (9)
网络通信 (5)
沙箱 (7)
内嵌及定制Lua引擎技术 (3)
IE控件及应用 (7)
反汇编 (15)
开源项目 (16)
C++ (15)
界面库 (3)
python (11)
疑难杂症 (24)
PHP (8)
Redis (8)
IT项目开发过程中的利器 (4)
libev源码解析 (6)

文章存档

2017年08月 (7)
2017年07月 (4)
2017年05月 (9)
2017年02月 (1)
2016年12月 (10)

展开

阅读排行

使用WinHttp接口实现HT (35595)
WMI技术介绍和应用—— (18359)
如何定制一款12306抢票: (13984)
一种准标准CSV格式的介 (12486)
一种精确从文本中提取UI (12203)
实现HTTP协议Get、Post: (11999)
分析两种Dump(崩溃日志 (11576)
一种解决运行程序报“应月 (11171)
实现HTTP协议Get、Post: (11158)
反汇编算法介绍和应用— (10676)

评论排行

使用WinHttp接口实现HT (33)
使用VC实现一个“智能”自 (27)
WMI技术介绍和应用—— (23)
WMI技术介绍和应用—— (20)
实现HTTP协议Get、Post: (20)
如何定制一款12306抢票: (17)
在windows程序中嵌入Lu (15)
一个分析“文件夹”选择框: (13)
反汇编算法介绍和应用— (12)
使用VC内嵌Python实现的 (10)

推荐文章

* CSDN日报20170817——《如果
不从事编程,我可以做什么?》

明了一个Foo对象。然后TEST_F(FRIEND_TEST_Test2, TEST_F)类直接使用了该成员变量调用Bar方法。
TEST_F(FRIEND_TEST_Test2, TEST_F)既继承于FRIEND_TEST_Test2, 又是Foo的友元类。

这块技术没有什么深奥的,大家只要是知道它是通过友元特性实现,是一种侵入式测试就行了。

顶 踩
0 0

上一篇 Google Test(GTest)使用方法和源码解析——自定义输出技术的分析和应用

下一篇 Google Test(GTest)使用方法和源码解析——参数自动填充技术分析和应用

相关文章推荐

- Google Test(GTest)使用方法和源码解析——死亡...
- Google Mock(Gmock)简单使用和源码分析
- 【直播】机器学习之凸优化--马博士
- 【套餐】2017软考系统集成项目管理工程师
- Google Test(GTest)使用方法和源码解析——结果...
- Google Test(GTest)使用方法和源码解析—
- 【直播】计算机视觉原理及实战--屈教授
- 【课程】深入探究Linux/VxWorks的设备树--牙
- Google Test(GTest)使用方法和源码解析——模板...
- Google Test(GTest)使用方法和源码解析——
- 机器学习&数据挖掘7周实训--韦玮
- 用google mock模拟C++对象
- Google Test(GTest)使用方法和源码解析——参数...
- python3编写简易统计服务器
- 机器学习之数学基础系列--AI100
- Google Test(GTest)使用方法和源码解析——自定...

查看评论

暂无评论

发表评论

用户名: GreatProgramer

评论内容:



提交

* 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

- * Android自定义EditText:你需要一款简单实用的SuperEditText(一键删除&自定义样式)
- * 从JDK源码角度看Integer
- * 微信小程序——智能小秘“遥知之”源码分享(语义理解基于olami)
- * 多线程中断机制
- * 做自由职业者是怎样的体验

最新评论

使用WinHttp接口实现HTTP协议(breaksoftware: @qq_34534425: 你过谦了。多总结、多练习、多借鉴就好了。

使用WinHttp接口实现HTTP协议(qq_34534425: 代码真心nb,感觉自己写的就是渣渣

朴素、Select、Poll和Epoll网络编程 zhangcunli8499: @Breaksoftware:多谢

朴素、Select、Poll和Epoll网络编程 breaksoftware: @zhangcunli8499:这篇 http://blog.csdn.net /breaksoftwa...

朴素、Select、Poll和Epoll网络编程 zhangcunli8499: 哥们,能传一下完整的代码吗?

C++拾趣——类构造函数的隐式\$ breaksoftware: @wuchalilun:多谢鼓励,其实我就想写出点不一样的地方,哈哈。

C++拾趣——类构造函数的隐式\$ Ray_Chang_988: 其他相关的explicit的介绍文章也看了,基本上explicit的作用也都解释清楚了,但是它们都没...

Redis源码解析——字典结构 breaksoftware: @u011548018:多谢鼓励

Redis源码解析——字典结构 生无可恋只能打怪升级: 就冲这图也得点1024个赞

WMI技术介绍和应用——查询系 breaksoftware: @hobbyonline:我认为这种属性的信息不准确是很正常的,因为它的正确与否不会影响到系统在不同...

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved

