

CoderZh的技术博客

一个程序员的思考与总结(请移步至：<http://blog.coderzh.com/>)

[博客园](#)[首页](#)[联系](#)[订阅](#)[管理](#)

公告

DigitalOcean优惠码

这里的博客将不再更新，最新博客

请移步至：

我的独立博客：

<http://blog.coderzh.com/>



微信公众号：

hacker-thinking

昵称：CoderZh

园龄：11年1个月

粉丝：790

关注：10

[+加关注](#)

搜索

找找看

随笔分类

[Agile\(2\)](#)

[Android\(3\)](#)

[ASP.NET\(3\)](#)

随笔-234 文章-10 评论-2047

玩转Google开源C++单元测试框架Google Test系列(gtest)之八 - 打造自己的单元测试框架

一、前言

上一篇我们分析了gtest的一些内部实现，总的来说整体的流程并不复杂。本篇我们就尝试编写一个精简版本的C++单元测试框架：nancyttest，通过编写这个简单的测试框架，将有助于我们理解gtest。

二、整体设计

使用最精简的设计，我们就用两个类，够简单吧：

1. TestCase类

包含单个测试案例的信息。

2. UnitTest类

负责所有测试案例的执行，管理。

三、TestCase类

TestCase类包含一个测试案例的基本信息，包括：测试案例名称，测试案例执行结果，同时还提供了测试案例执行的方法。我们编写的测试案例都继承自TestCase类。



```
class TestCase
{
public:
    TestCase(const char* case_name) : testcase_name(case_name){}
```

[C#\(20\)](#)
[C/C++\(24\)](#)
[Cocos2d-x\(1\)](#)
[Emacs\(2\)](#)
[Google App Engine\(7\)](#)
[JAVA\(3\)](#)
[Linux\(1\)](#)
[Lua\(2\)](#)
[Python\(66\)](#)
[Ubuntu\(9\)](#)
[VBS\(4\)](#)
[安全性测试\(9\)](#)
[测试生活感悟\(7\)](#)
[程序人生\(15\)](#)
[代码安全\(3\)](#)
[单元测试\(19\)](#)
[公告\(13\)](#)
[每周总结\(4\)](#)
[软件测试\(30\)](#)
[设计模式](#)
[性能测试\(7\)](#)
[学习笔记\(27\)](#)

随笔档案

[2015年9月 \(1\)](#)
[2015年8月 \(2\)](#)
[2015年6月 \(4\)](#)
[2015年5月 \(2\)](#)
[2015年4月 \(5\)](#)
[2015年3月 \(1\)](#)

```
// 执行测试案例的方法
virtual void Run() = 0;

int nTestResult; // 测试案例的执行结果
const char* testcase_name; // 测试案例名称
};
```

四、UnitTest类

我们的UnitTest类和gtest的一样，是一个单件。我们的UnitTest类的逻辑非常简单：

1. 整个进程空间保存一个UnitTest 的单例。
2. 通过RegisterTestCase()将测试案例添加到测试案例集合testcases_中。
3. 执行测试案例时，调用UnitTest::Run()，遍历测试案例集合testcases_，调用案例的Run()方法

```
class UnitTest
{
public:
    // 获取单例
    static UnitTest* GetInstance();

    // 注册测试案例
    TestCase* RegisterTestCase(TestCase* testcase);

    // 执行单元测试
    int Run();

    TestCase* CurrentTestCase; // 记录当前执行的测试案例
};
```

2014年5月 (2)
 2014年4月 (2)
 2011年5月 (1)
 2011年3月 (1)
 2011年1月 (1)
 2010年12月 (3)
 2010年11月 (3)
 2010年10月 (2)
 2010年9月 (6)
 2010年8月 (2)
 2010年7月 (4)
 2010年6月 (3)
 2010年5月 (4)
 2010年4月 (9)
 2010年3月 (6)
 2010年2月 (3)
 2010年1月 (16)
 2009年12月 (6)
 2009年11月 (3)
 2009年10月 (4)
 2009年9月 (3)
 2009年8月 (2)
 2009年7月 (7)
 2009年6月 (2)
 2009年4月 (12)
 2009年3月 (5)
 2009年2月 (2)
 2009年1月 (3)
 2008年12月 (7)

```
int nTestResult; // 总的执行结果
int nPassed; // 通过案例数
int nFailed; // 失败案例数
protected:
    std::vector<TestCase*> testcases_; // 案例集合
};
```

下面是UnitTest类的实现：

```
UnitTest* UnitTest::GetInstance()
{
    static UnitTest instance;
    return &instance;
}

TestCase* UnitTest::RegisterTestCase(TestCase* testcase)
{
    testcases_.push_back(testcase);
    return testcase;
}

int UnitTest::Run()
{
    nTestResult = 1;
    for (std::vector<TestCase*>::iterator it = testcases_.begin();
         it != testcases_.end(); ++it)
    {
        TestCase* testcase = *it;
        CurrentTestCase = testcase;
        std::cout << green << "======" << std::endl;
        std::cout << green << "Run TestCase:" << testcase->testcase_name << std::endl;
```

[2008年11月 \(9\)](#)[2008年9月 \(8\)](#)[2008年8月 \(7\)](#)[2008年7月 \(8\)](#)[2008年6月 \(9\)](#)[2008年5月 \(33\)](#)[2008年4月 \(6\)](#)[2008年2月 \(1\)](#)[2007年12月 \(3\)](#)[2007年11月 \(3\)](#)[2007年10月 \(7\)](#)[2007年9月 \(1\)](#)

系列文章

[Python天天美味系列](#)[攻击方式学习系列](#)[瘦客户端那些事](#)[玩转gtest系列](#)

读书笔记

[Python网络编程](#)[xUnit Test Patterns](#)[卓有成效的程序员](#)

友情链接

积分与排名

积分 - 547334

排名 - 200

最新评论

```

        testcase->Run();
        std::cout << green << "End TestCase:" << testcase->testcase_name << std::endl;
        if (testcase->nTestResult)
        {
            nPassed++;
        }
        else
        {
            nFailed++;
            nTestResult = 0;
        }
    }

    std::cout << green << "=====" << std::endl;
    std::cout << green << "Total TestCase : " << nPassed + nFailed << std::endl;
    std::cout << green << "Passed : " << nPassed << std::endl;
    std::cout << red << "Failed : " << nFailed << std::endl;
    return nTestResult;
}

```



五、NTEST宏

接下来定一个宏NTEST，方便我们写我们的测试案例的类。



```

#define TESTCASE_NAME(testcase_name) \
    testcase_name##_TEST

#define NANCY_TEST_(testcase_name) \
class TESTCASE_NAME(testcase_name) : public TestCase \
{ \
public: \

```

1. Re:gtest参数化测试代码示例

博客园的链接改了，是这个地址：

--canbeing

2. Re:玩转Google开源C++单元测试框架Google Test系列(gtest)之一 - 初识gtest

@xiao_1bai编译的目标就是生成lib文件，你已经成功了。现在可以在你的项目引用gtestd.lib...

--cnbloghzc

3. Re:ViEmuVS2013-3.2.1 破解

安装失败，提示：

所需要的.NET Framework 没有

--xiake007

4. Re:玩转Google开源C++单元测试框架Google Test系列(gtest)之七 - 深入解析gtest

谢谢，作者哥哥，这么多章，最精彩这章。领教了，谢谢

谢谢，作者哥哥，这么多章，最精彩这章。领教了，谢谢

--\$JackChen

5. Re:最常用的Emacs的基本操作

如果Emacs入手都算有些难度。。那VIM怎么办？

--震灵

6. Re:PyQt4学习资料汇总

大神，我下了你的财务管理系统，那个数据库文件提示打不开，怎么解决啊~~急用~~

--lzgst

```
TESTCASE_NAME(testcase_name)(const char* case_name) : TestCase(case_name){}; \
virtual void Run(); \
private: \
    static TestCase* const testcase_; \
}; \
\
TestCase* const TESTCASE_NAME(testcase_name) \
::testcase_ = UnitTest::GetInstance()->RegisterTestCase( \
    new TESTCASE_NAME(testcase_name)(#testcase_name)); \
void TESTCASE_NAME(testcase_name)::Run()

#define NTEST(testcase_name) \
    NANCY_TEST_(testcase_name)
```



六、RUN_ALL_TEST宏

然后是执行所有测试案例的一个宏：

```
#define RUN_ALL_TESTS() \
    UnitTest::GetInstance()->Run();
```

七、断言的宏EXPECT_EQ

这里，我只写一个简单的EXPECT_EQ：



```
#define EXPECT_EQ(m, n) \
    if (m != n) \
    { \
        UnitTest::GetInstance()->CurrentTestCase->nTestResult = 0; \
        std::cout << red << "Failed" << std::endl; \
```

7. Re:ViEmuVS2013-3.2.1 破解

为啥我的注册表中没有whole

tomato

--蓝域小兵

8. Re:玩转Google开源C++单元测试框架Google Test系列(gtest)之三 - 事件机制

由于没有加TEST 宏，输出结果如下：[=====] Running 0 tests from 0 test cases.[=====] 0 tests from 0 test c.....

--喜马拉雅

9. Re:从CEGUI源码看代码规范

好一个singleton！

--chaosink

10. Re:使用UI Automation库用于UI自动化测试

mark

--大恒爸爸

阅读排行榜

1. 玩转Google开源C++单元测试框架Google Test系列(gtest)(总)(221480)
2. 玩转Google开源C++单元测试框架Google Test系列(gtest)之一 - 初识gtest(146539)
3. 玩转Google开源C++单元测试框架Google Test系列(gtest)之二 - 断言(105010)

```
std::cout << red << "Expect:" << m << std::endl; \
std::cout << red << "Actual:" << n << std::endl; \
}
```



八、案例Demo

够简单吧，再来看看案例怎么写：



```
#include "nancytest.h"

int Foo(int a, int b)
{
    return a + b;
}

NTEST(FooTest_PassDemo)
{
    EXPECT_EQ(3, Foo(1, 2));
    EXPECT_EQ(2, Foo(1, 1));
}

NTEST(FooTest_FailDemo)
{
    EXPECT_EQ(4, Foo(1, 2));
    EXPECT_EQ(2, Foo(1, 2));
}

int _tmain(int argc, _TCHAR* argv[])
{
```

4. 玩转Google开源C++单元测试框架Google Test系列(gtest)之三 - 事件机制(68517)
5. 玩转Google开源C++单元测试框架Google Test系列(gtest)之六 - 运行参数(54532)
6. 玩转Google开源C++单元测试框架Google Test系列(gtest)之四 - 参数化(54400)
7. C# 中使用JSON - DataContractJsonSerializer(52402)
8. PyQt4学习资料汇总(49575)
9. 玩转Google开源C++单元测试框架Google Test系列(gtest)之七 - 深入解析gtest(49311)
10. 代码覆盖率浅谈(47554)

评论排行榜

1. PyQt4学习资料汇总(152)
2. 开源Granados介绍 - SSH连接远程Linux服务器(C#)(66)
3. (原创)攻击方式学习之(1) - 跨站式脚本(Cross-Site Scripting) (49)
4. 三年之痒(44)
5. NancyBlog - 我的Google App Engine Blog(42)
6. 创业三年来的一些感想 - 游戏篇(40)
7. CCNET+MSBuild+SVN实时构建的优化总结(40)

```
return RUN_ALL_TESTS();  
}
```

整个一山寨版gtest，呵。执行一下，看看结果怎么样：



```
=====  
Run TestCase:FooTest_PassDemo  
End TestCase:FooTest_PassDemo  
=====  
Run TestCase:FooTest_FailDemo  
Failed  
Expect:4  
Actual:3  
Failed  
Expect:2  
Actual:3  
End TestCase:FooTest_FailDemo  
=====  
Total TestCase : 2  
Passed : 1  
Failed : 1  
Press any key to continue . . .
```

九、总结

本篇介绍性的文字比较少，主要是我们在上一篇深入解析gtest时已经将整个流程弄清楚了，而现在编写的nancytest又是其非常的精简版本，所有直接看代码就可以完全理解。希望通过这个Demo，能够让大家对gtest有更加直观的了解。回到开篇时所说的，我们没有必要每个人都造一个轮子，因为gtest已经非常出色的为我们做好了这一切。如果我们每个人都写一个自己的框架的话，一方面我们要付出大量的维护成本，一方面，这个框架也许只能对你有用，无法让大家从中受益。

8. CoderZh首款Python联机对战游戏 - NancyTetris1.0倾情发布 (一) (37)
9. 代码安全系列(1) - Log的注入(35)
10. 程序员的信仰(35)

推荐排行榜

1. 玩转Google开源C++单元测试框架Google Test系列(gtest)(总)(24)
2. 创业三年来的一些感想 - 游戏篇 (14)
3. 程序员的共鸣 - 读《卓有成效的程序员》(12)
4. 代码覆盖率浅谈(12)
5. 《xUnit Test Patterns》学习笔记 5 - xUnit基础(10)
6. 三年之痒(9)
7. 玩转Google开源C++单元测试框架Google Test系列(gtest)之一 - 初识gtest(9)
8. 优美的测试代码 - 行为驱动开发(BDD)(8)
9. Python天天美味(总)(7)
10. PyQt4学习资料汇总(6)

gtest正是这么一个优秀C++单元测试框架，它完全开源，允许我们一起为其贡献力量，并能让更多人从中受益。如果你在使用gtest过程中发现gtest不能满足你的需求时（或发现BUG），gtest的开发人员非常急切的想知道他们哪来没做好，或者是gtest其实有这个功能，但是很多用户都不知道。所以你可以直接联系gtest的开发人员，或者你直接在这里回帖，我会将您的意见转告给gtest的主要开发人员。如果你是gtest的超级粉丝，原意为gtest贡献代码的话，加入他们吧。

本Demo代码下载：</Files/coderzh/Code/nancytest.rar>

本篇是该系列最后一篇，其实gtest还有更多东西值得我们去探索，本系列也不可能将gtest介绍完全，还是那句话，想了解更多gtest相关的内容的话：

访问官方主页：<http://code.google.com/p/googletest/>

下载gtest源码：<http://code.google.com/p/googletest/downloads/list>

系列链接：

- 1.玩转Google开源C++单元测试框架Google Test系列(gtest)之一 - 初识gtest
- 2.玩转Google开源C++单元测试框架Google Test系列(gtest)之二 - 断言
- 3.玩转Google开源C++单元测试框架Google Test系列(gtest)之三 - 事件机制
- 4.玩转Google开源C++单元测试框架Google Test系列(gtest)之四 - 参数化
- 5.玩转Google开源C++单元测试框架Google Test系列(gtest)之五 - 死亡测试
- 6.玩转Google开源C++单元测试框架Google Test系列(gtest)之六 - 运行参数
- 7.玩转Google开源C++单元测试框架Google Test系列(gtest)之七 - 深入解析gtest
- 8.玩转Google开源C++单元测试框架Google Test系列(gtest)之八 - 打造自己的单元测试框架



DigitalOcean的VPS主机，稳定、速度快、价格也实惠。可以在上面部署独立网站或各种实用工具。

我用了很久了，确实不错，极力推荐。

使用这个链接购买可获得10美元优惠。

优惠链接：[DigitalOcean优惠码](#)



微信扫一扫交流

作者：[CoderZh](#)

公众号：hacker-thinking（一个程序员的思考）

独立博客：<http://blog.coderzh.com>

博客园博客将不再更新，请关注我的「微信公众号」或「独立博客」。

作为一个程序员，思考程序的每一行代码，思考生活的每一个细节，思考人生的每一种可能。

文章版权归本人所有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，否则保留追究法律责任的权利。

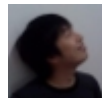
分类: [单元测试, C/C++](#)

标签: [Google Test](#)

好文要顶

关注我

收藏该文



CoderZh

关注 - 10

粉丝 - 790

+加关注

5

0

« 上一篇：[试用了Eric4，打算在Eric4中使用Pyqt4写个GUI程序](#)

» 下一篇：[程序员的信仰](#)

评论列表

#1楼 2009-05-15 20:34 haonan

Gtest简直太强大了，顶礼膜拜啊~~

支持(0) 反对(0)

#2楼 2010-05-11 16:48 xavier1

good

#3楼 2010-05-11 16:51 xavier1

支持下楼主：

有2个问题：

1、lz能否介绍一下在不同平台下如何编译搭建框架。xp/winCE/symbian，对了是否可以在Android下使用这个框架？

2 目前gtest相关的资料比较少，可否msn交流，官网目前无法提问。

我的msn：magic_xavier@hotmail.com

期盼回复。

#4楼[楼主] 2010-05-12 08:45 CoderZh

@ xavier1

谢谢你的支持。

Google's framework for writing C++ tests on a variety of platforms (Linux, Mac OS X, Windows, Cygwin, Windows CE, and Symbian)

所以，xp/WinCE/Symbian是没有问题的，Android是基于Java，应该不支持。Android本身已经自带了测试框架

android.test，应该就足够了。

msn已加

支持(0) 反对(0)

#5楼[楼主] 2010-05-12 08:46 CoderZh

@ xavier1

wince里使用参考：

<http://rxwen.blogspot.com/2010/05/use-googletest-on-windows-ce.html>

不过要先翻墙

支持(0) 反对(0)

#6楼 2010-05-12 10:36 xavier1

@ CoderZh

谢谢，你给的wince参考我连接不了，怎么翻墙？

可否邮件发给我。

我已经把gtest针对pocket pc2003的库编译过去了。

但是winCE上面运行console就是什么也看不到。gtest是支持console和xml输出的。在winCE上怎么才能看到输出结果？

另：msn消息发送不成功

#7楼 2012-02-21 11:30 zampoo

挺好的 挺好的

支持(0) 反对(0)

#8楼 2016-03-16 14:53 海天黄豆将

请问 gtest能用来进行多线程测试吗，如何进行？

支持(0) 反对(0)

#9楼 2016-06-17 15:27 稀有品种

您好！

请教本Demo代码下载：/Files/coderzh/Code/nancytest.rar

这个地址是哪里？

支持(0) 反对(0)

#10楼 2016-12-07 15:52 超超boy

前来膜拜。。。

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】腾讯云上实验室 1小时搭建人工智能应用

【推荐】可嵌入您系统的“在线Excel”！SpreadJS 纯前端表格控件

【推荐】阿里云“全民云计算”优惠升级



最新IT新闻:

- T-Mobile美国和Sprint正商讨合并 未来数周出结果
 - iPhone芯片和屏幕皆出自他手 三星扼住苹果咽喉
 - 贾跃亭已偿还债务200亿元 乐视称将分批还债
 - iOS 11正式版发布：可以升级了！
 - 国行iPhone 8/8 Plus首发开箱！双玻璃果然漂亮
- » 更多新闻...



最新知识库文章:

- Google 及其云智慧
 - 做到这一点，你也可以成为优秀的程序员
 - 写给立志做码农的大学生
 - 架构腐化之谜
 - 学会思考，而不只是编程
- » 更多知识库文章...