

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

编写优美的GTest测试案例

使用gtest也有很长一段时间了，这期间也积累了一些经验，所以分享一下。GTest为我们提供了便捷的测试框架，让我们只需要关注案例本身。如何在GTest框架下写出优美的测试案例，我觉得必须要做到：

1. 案例的层次结构一定要清晰
2. 案例的检查点一定要明确
3. 案例失败时一定要能精确的定位问题
4. 案例执行结果一定要稳定
5. 案例执行的时间一定不能太长
6. 案例一定不能对测试环境造成破坏
7. 案例一定独立，不能与其他案例有先后关系的依赖
8. 案例的命名一定清晰，容易理解

案例的可维护性也是非常重要，如果做到上面的8点，自然也就做到了可维护性。下面来分享一下我对于上面8点的经验：

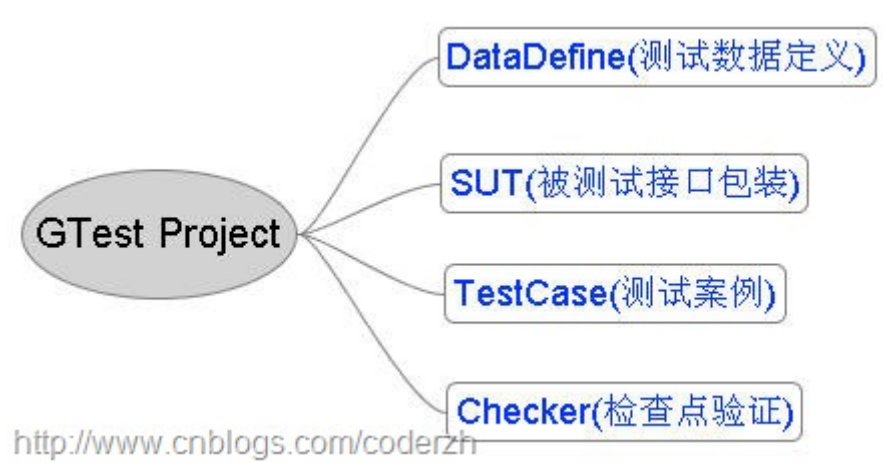
1. 案例的层次结构一定要清晰

所谓层次结构，至少要让人一眼就能分辨出被测代码和测试代码。简单的说，就是知道你在测什么。由于是进行接口测试，我已经习惯了如下的案例层次：

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)



DataDefine

我会将测试案例所需要的数据，以及数据之间的联系全部在预先定义好。测试数据与案例逻辑的分离，有利于维护和扩展测试案例。同时，GTest先天就支持测试数据参数化，为测试数据的分离提供了进一步的便捷。什么是测试数据参数化？就是你可以预先定义好一批各种各样的数据，而你只需要编写一个测试案例的逻辑代码，gtest会将定义好的数据逐个套入测试案例中进行执行。具体的做法请见：[玩转Google开源C++单元测试框架Google Test系列\(gtest\)之四 - 参数化](#)

SUT

SUT，即system under test，表明你的测试对象是什么，它可以是一个类(CUT)，对象(OUT)，函数(MUT)，甚至可以是整个应用程序(AUT)。我单独将这个层次划分出来，主要有两个目的：

- 明确的表示出你的测试对象是什么
- 为复杂调用对象包装简单调用接口

明确表示测试对象是什么，便于之后对测试案例的维护和对测试案例的理解。同时，对于一些被测对象，你想要调用它需要经过一系列烦琐的过程，这时，就需要将这一烦琐的调用过程隐藏起来，而只关注被测对象的输入和输出。

TestCase

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)

测试工程中，必须非常明确的表示出哪些是测试案例，哪些是其他的辅助文件。通常，我们会在测试案例的文件名加上Test前缀(或者后缀)。我建议，将所有的测试案例文件或代码放在最显眼的地方，让所有看到你的测试工程的人，第一眼看到的就是测试案例，这很重要。

Checker

对于一个复杂系统的接口测试，仅仅坚持输入和输出是远远不够的。比如测试一个写数据库的函数，函数的返回值告诉你数据已经成功写入是远远不够的，你必须亲身去数据库中查个究竟才行。因此，对于某一类的测试案例，我们可以抽象出一些通用的检查点代码。

如果做到上面的分层，那么一个测试案例写出来的结构应该会是这个样子：



```
TEST(TestFoo, JustDemo)
{
    GetTestData(); // 获取测试数据

    CallSUT(); // 调用被测方法

    CheckSomething(); // 检查点验证
}
```



这样的测试案例，一目了然。

2. 案例的检查点一定要明确

一定要明确案例的检查点是什么，并且让检查点尽量集中。有一个不好的习惯就是核心的检查点在分布在多个函数中，需要不断的跳转才能了解到这个案例检查了些什么。好的做法应该是尽量让检查点集中，能够非常清晰的分辨出案例对被测代码做了哪些检查。所以，尽量让Gtest的ASSERT_和EXPECT_系列的宏放在明显和正确的地方。

3. 案例失败时一定要能精确的定位问题

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

测试案例失败时，我们通常手忙脚乱。如果一个测试案例Failed，却不能立即推断是被测代码的Bug的话，这个测试案例也有待改进。我们可以在一些复杂的检查点断言中加入一些辅助信息，方便我们定位问题。比如下面这个测试案例：

```
int n = -1;
bool actualResult = Foo::Dosomething(n);
ASSERT_TRUE(actualResult)
```

如果测试案例失败了，会得到下面的信息：

```
Value of: actualResult
Actual: false
Expected: true
```

这样的结果对于我们来说，几乎没有什么用。因为我们根本不知道actualResult是什么，以及在什么情况下才会出现非预期值。因此，在断言处多加入一些信息，将有助于定位问题：

```
int n = -1;
bool actualResult = Foo::Dosomething(n);
ASSERT_TRUE(actualResult) << L"Call Foo::Dosomething(n) when n = " << n;
```

4. 案例执行结果一定要稳定

要保证测试案例在什么时候、什么情况下执行的结果都是一样的。一个一会成功一会失败的案例是没有意义的。要保证案例稳定性的方法有很多，比如杜绝案例之间的影响，有时候，由于前一个案例执行完后，将一些系统的环境破坏了，导致后面的案例执行失败。在测试某些本身就存在一定几率或延时的系统时，使用超时机制是比较简单的办法。比如，你需要测试一个启动Windows服务的方法，如果我们在调用了该方法后立即进行检查，很可能检查点会失败，有时候也许又是通过的。这是因为Windows服务由Stop状态到Running状态，中间还要经过一个Padding状态。所以，简单的做法是使用超时机制，隔断时间检查一次，直到超过某个最大忍受时间。



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

--\$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

```
ASSERT_TRUE(StartService('xxx'));
int tryTimes = 0;
int status = GetServiceStatus('xxx');
while (status != Running)
{
    if (tryTimes >= 10)
        break;
    ::Sleep(200);
    tryTimes++;
    status = GetServiceStatus('xxx');
}
ASSERT_EQ(Running, status) << "Check the status after StartService('xxx')";
```



5. 案例执行的时间一定不能太长

我们应该尽量让案例能够快速的执行，一方面，我们可以通过优化我们的代码来减少运行时间，比如，减少对重复内容的读取。一方面，对于一些比较耗时的操作，比如文件系统，网络操作，我们可以使用Mock对象来替代真实的对象。使用GMock是一个不错的选择。

6. 案例一定不能对测试环境造成破坏

有的案例需要在特定的环境下来能执行，因此会在案例的初始化时对环境进行一些修改。注意，不管对什么东西进行了修改，一定要保证在案例执行完成的TearDown中将这环境都还原回来。否则有可能对后面的案例造成影响，或者出现一些莫名其妙的错误。

7. 案例一定独立，不能与其他案例有先后关系的依赖

任何一个案例都不依赖于其他测试案例，任何一个案例的执行结果都不应该影响到别的案例。任何一个案例都可以单独拿出来正确的执行。所以，不能寄希望于前一个案例所做的环境准备，因为这是不对的。

8. 案例的命名一定清晰，容易理解

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

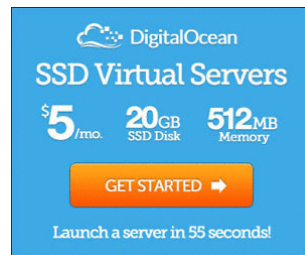
1. 玩转Google开源C++单元测试框架Google Test系列(gtest)(总)(221480)
2. 玩转Google开源C++单元测试框架Google Test系列(gtest)之一 - 初识gtest(146539)
3. 玩转Google开源C++单元测试框架Google Test系列(gtest)之二 - 断言(105010)

案例的名字要规范，长不要紧，一定要清晰的表达测试案例的用途。比如，下面的测试案例名称都是不好的：

```
TEST(TestFoo, Test)
TEST(TestFoo, Normal)
TEST(TestFoo, Alright)
```

比如像下面的案例名称就会好一点：

```
TEST(TestFoo, Return_True_When_ParameterN_Larger_Then_Zero)
TEST(TestFoo, Return_False_When_ParameterN_Is_Zero)
```



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。

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)

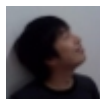
分类: [C/C++](#), [软件测试](#)

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

« 上一篇: [时间管理GTD工具](#)

» 下一篇: [Python查询Google PageRank](#)

3

0

posted @ 2010-01-09 14:26 CoderZh 阅读(22383) 评论(3) 编辑 收藏

评论列表

#1楼 2010-04-13 18:16 luowenming

请教下楼主：

我要对现有的一个系统用gtest进行单元测试，但系统中类与类相互有依赖关系。而且可能嵌套会很深，这时用gtest进行测试时，是不是需要先去把它们之间的依赖关系理清？实在没招了呀。

如果是这样的话，就有可能有这种情况：我只需要测试一个类中的某一个函数，这个类和其它很多个类都有关系，这就需要花大量时间去研究。给点建议吧，谢谢

支持(0) 反对(0)

#2楼[楼主] 2010-04-13 20:51 CoderZh

@ luowenming

首先，如果能够对系统进行重构，你可以参与进去，提一些可测试性意见。

其次，尝试使用mock替代依赖对象。可参考gmock框架。

第三，复杂系统难免会遇到你说的问题，也比较正常。找到最小的合适的粒度，进行单元测试。同时，从最外层进行测试保证系统整体功能也是很有必要的。所以，花些时间了解清楚内部类与类之间的关系吧。

支持(0) 反对(0)

#3楼 2016-11-16 09:11 beichen2012

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)

请教下楼主：

我是想测试安卓NDK里面的opensles部分代码（录音播放），感觉无从下手。给点建议吧，谢谢。

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

美团云 MERCHANT OPEN SERVICES

GPU 云主机 5折起

M60 原价1660/月 现价830/月 P40 原价3400/月 现价2100/月

[查看详情](#)

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...](#)

站长统计