



新闻

体育

汽车

房产

旅游

教育

时尚

科技

财经

娱乐

更多

登录狐友

0

分享到

腾讯移动品质
中心TMQ105
文章 | 2.1万
总阅读

查看TA的文章>

基于hook和gmock开展单元测试

2017-06-15 12:14

程序设计 / 谷歌 / 操作系统

一、什么是UT

单元测试（英语：Unit Testing）又称为模块测试，是针对程序模块（软件设计的最小单位）来进行正确性检验的测试工作。程序单元是应用的最小可测试部件。在过程化编程中，一个单元就是单个程序、函数、过程等。

对于面向对象编程，最小单元就是方法，包括基类（超类）、抽象类、或者派生类（子类）中的方法——摘自维基百科。

二、为什么要做UT

16年下半年对滴滴SDK接口进行梳理，并进行了BVT接口自动化以及截图半自动化效果验证，但是有几个问题没能得到很好的解决：

（1）SDK的整体代码行覆盖率是57.6%，但导航引擎的覆盖率仅31.2%；

（2）从SDK这层测试导航引擎，需要回放不同类型的轨迹，测试效率低；

（3）从端上直接测试引擎，不符合分层测试思想，较难发现深层次问题。

大家都在搜：iPhone

只敢卖1天
货到付款

热门图集



24小时热文

1

最美“水
恋爱了

2

人工智
情内容

3

华硕Zer
820处理武警也
TFBOY郭明錤
解决供

0

分享到

(1) 熟悉被测模块

无论是做自动化测试也好，集成测试也罢，都需要对待测模块有一定程度的了解，对于单元测试这种需要深入代码逻辑的测试来讲，更是如此。在开展测试之前，主要从几个方面对待测模块进行分析：代码逻辑、圈复杂度、代码深度、扇入、扇出以及代码行等，如下图1所示：

function	effect	Complexity	Statements	Max Depth	Calls
funA	GPS点匹配	228	416	7	224
funB	返回是否是有效点	137	262	3	0
funC	初始化接口	130	289	7	157
funD	设置气泡状态	115	307	8	237
funE	获取下一个事件点	13	19	3	9
funF	寻找最近点	6	6	4	0
funG	获取地图状态	5	6	4	0

图1可测性分析

可以看到，该模块有些接口的圈复杂度达到了200+，而业内设计较好的代码圈复杂度在15左右，对这类接口，不建议做UT，最好的方法是让开发进行优化，降低函数的圈复杂度。

（2）选用合适的测试框架

工欲善其事必先利其器，对UT而言也是如此。C++的历史已经非常悠久了，开源框架也是非常多，其中google公司出品的gtest和gmock就是做C++单测的必备神器

(<https://github.com/google/googletest>)。



这所高校只有女生 她们要毕业了

24小时热文

1

最美“水
恋爱了

2

人工智能
情内容

3

华硕Zen
820处理



TFBOYS



郭明錫
解決供應

24小时热文

1

华为亮出
P11将是iPhoneX终结者

2

越南路边“洗碗小妹”抬起头后惊艳路人

3

华硕Zenfone V发布：搭载骁龙820处理器



武警也来排队，OPPO R11
TFBOYS限量版太火了吧



新闻

体育

汽车

房产

旅游

教育

时尚

科技

财经

娱乐

更多

登录狐友

0

结合SDK实际情况，整合gtest和gmock框架至测试分支，如下图2所示：

分享到

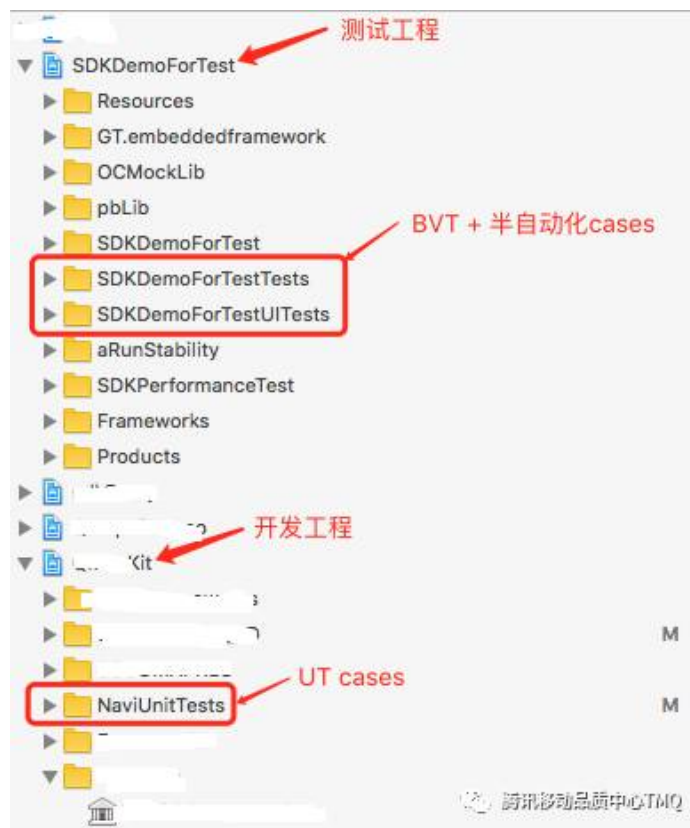


图2代码组织结构

这里的UT是嵌入到开发工程里的，做为开发源码WorkSpace中的一个target，该target和之前BVT的target的区别在于，其是基于MAC OSX的Command Line工程，运行环境是MAC



解决供应不足问题

24小时热文



1

最美“水
恋爱了

2

人工智
情内容

3

华硕Zei
820处理

搜狐号推荐



IT之家
IT之家是业内领先
网站。IT之家快速



郭明錤
解决供



搜狐科技视界
搜狐科技官方原创...
件、大趋势和新变化，用我们的视角观...



懂懂笔记
●业内资深财经记者 ●长年跟踪报道IT、互联
网、电信领域，近二十年的积累，...



最黑科技
带你认识全球最黑科技！



雷锋网
雷锋网是为数不多的前沿行业垂直媒体，曾经
是“硬件复兴”、“创客运动”、...



新闻

体育

汽车

房产

旅游

教育

时尚

科技

财经

娱乐

更多

登录狐友

iOS Simulator系统，这些差别所带来的影响会在第4节中详细说明。

0

分享到

(3) 设计单测case

环境部署好了，剩下的就是根据之前的接口分析来设计单测case了。这里举一个简单的例子来进行说明，被测接口是getItem，代码逻辑比较简单，如下图3所示：

```
CQRouteFor *getItem(int i){  
    if (NULL == m_pItem[0]) {  
        for (int i = 0; i < 3; i++) {  
            m_pItem[i] = new CQRouteFor;  
        }  
    }  
    if (i >= 0 && i <= 3) {  
        return m_pItem[i];  
    }  
    return NULL;  
}
```

腾讯移动品质中心TMQ

图3被测接口

如何设计case呢？对这种既有入参，又有返回值的函数，相对是比较好设计case并进行结果验证的，我们重点关注入参i在不同取值的情况下，函数返回结果是否符合预期。测试代码的编写如下图4：

晚上可以看

24小时热文

1

最美“水
恋爱了

2

人工智
情内容

3

华硕Zel
820处理

联系我们



武警也
TFBOY



郭明錤
解决供



新闻

体育

汽车

房产

旅游

教育

时尚

科技

财经

娱乐

更多

登录狐友

0

分享到

```
class CQRouteForTest : public testing::Test{
protected:
    CQRouteFor *routeFor;
    virtual void SetUp() {
        routeFor = new CQRouteFor;
    }

    virtual void TearDown() {
        free(routeFor);
    }
};

#pragma mark -- 测试用例
TEST_F(CQRouteForTest, test_getItem){
    //正常逻辑
    routeFor->m_pItem[0] = NULL;
    EXPECT_NE(NULL, routeFor->getItem(2));
    routeFor->m_pItem[0] = new CQRouteFor;
    EXPECT_EQ(NULL, routeFor->getItem(5));
    //边界值
    EXPECT_NE(NULL, routeFor->getItem(0));
    EXPECT_NE(NULL, routeFor->getItem(3));
    //异常值
    EXPECT_EQ(NULL, routeFor->getItem(-1));
    EXPECT_EQ(NULL, routeFor->getItem(100));
}
```

图4测试用例

这样的case是不是很简单，但在写单测的过程中，我们所面对的测试对象往往复杂的超出你的想象。

四、遇到的问题与解决方案

24小时热文

1

最美“水
恋爱了

2

人工智
情内容

3

华硕Ze
820处理武警也
TFBOY郭明錤
解决供



新闻

体育

汽车

房产

旅游

教育

时尚

科技

财经

娱乐

更多

登录狐友

0

分享到

开发在设计类时，对于不想让外部类访问的属性以及方法都可以定义为私有的，这并没有什么设计上的问题，但对于测试而言，就要突破这种访问限制，做到public和非public接口都可以在测试类中被访问到，对这个问题，最简洁快速的方法是：在测试类中将private、protected关键字重定义为public，之后在测试类中就可以访问到被测函数的所有方法以及属性。代码如下图5：

```
#define private public
#define protected public

#include "QRouteGuidance.h"
#include "QRouteGuidanceApi.h"
#include "QRouteGuidanceTextPool.h"

#undef private
#undef protected
```

腾讯移动品质中心TMQ

图5private可访问

（2）对回调函数的测试

对于C++中的异步回调，可以采用异步变同步的方法，保证该调的时候可以正常的调用。

（3）static以及非虚函数，无法使用现有的框架进行mock

1）为什么无法mock static类型的函数？

24小时热文

1

最美“水
恋爱了

2

人工智
情内容

3

华硕Zer
820处理



武警也
TFBOY



郭明錤
解决供



新闻

体育

汽车

房产

旅游

教育

时尚

科技

财经

娱乐

更多

登录狐友

0

分享到

make some changes.即如果你需要mock一个静态函数，那说明你的程序模块过于“紧耦合”了（并且灵活性不够、重用性不够、可测试性不够），你最好是定义一个小接口，通过这个接口来调用那个函数，然后就容易mock了。

2) 为什么无法mock非虚函数？

C++ allows a subclass to change the access level of a virtual function in the base class.

C++允许用基类的指针来调用子类的函数，举个例子，就很容易明白了，如图6：

```
class A
{
public:
    virtual void foo()
    {
        cout<<"A::foo() is called"<<endl;
    }
};
class B:public A    在使用mock时，定义的mock类要继承被测类
{
public:
    void foo()
    {
        cout<<"B::foo() is called"<<endl;
    }
};
int main(void)
{
    A *a = new B();
    a->foo();    // 在这里，a虽然是指向A的指针，但是被调用的函数(foo)却是B的！
    return 0;
}
```

腾讯移动品质中心TMQ

图6基类指针调子类函数

24小时热文

1

最美“水
恋爱了

2

人工智
情内容

3

华硕Zer
820处理



武警也
TFBOY



郭明錤
解决供

0

分享到

所有的接口都定义为虚函数，那这个问题如何解呢？

方案一

见 google官方手册
<https://github.com/google/googletest/blob/master/googlemock/docs/CookBook.md> ,
Google Mock can mock non-virtual functions to be used in what we call hi-perf
dependency injection，即依赖注入。该方案的原理是通过模板类的方式来实现，在开发代
码中通过传入实际对象来调用真实接口，在测试代码中通过传入mock对象来调用mock出来
的接口。Google官方提供的一个例子，如图7：

- 24小时热文
- 1

最美“水
恋爱了
- 2

人工智
情内容
- 3

华硕Zei
820处理
- 

武警也
TFBOY
- 

郭明錤
解决供



新闻

体育

汽车

房产

旅游

教育

时尚

科技

财经

娱乐

更多

登录狐友

0

分享到

```

class ConcretePacketStream {
public:
    void AppendPacket(Packet* new_packet);
    const Packet* GetPacket(size_t packet_number) const;
    size_t NumberOfPackets() const;
};

// A mock packet stream class. It inherits from no other, but defines
// GetPacket() and NumberOfPackets().
class MockPacketStream {
public:
    MOCK_CONST_METHOD1(GetPacket, const Packet*(size_t packet_number));
    MOCK_CONST_METHOD0(NumberOfPackets, size_t());
};

template <class PacketStream>
void CreateConnection(PacketStream* stream) { ... }

template <class PacketStream>
class PacketReader {
public:
    void ReadPackets(PacketStream* stream, size_t packet_num);
};

CreateConnection<ConcretePacketStream>()
PacketReader<ConcretePacketStream>

CreateConnection<MockPacketStream>()
PacketReader<MockPacketStream>

MockPacketStream mock_stream;
EXPECT_CALL(mock_stream, ...)...;
.. set more expectations on mock_stream ...
PacketReader<MockPacketStream> reader(&mock_stream);
.. exercise reader ...

```

开发代码

测试代码

24小时热文

1

最美“水
恋爱了

2

人工智
情内容

3

华硕Zer
820处理武警也
TFBOY郭明錤
解决供

图7 依赖注入

方案二



新闻

体育

汽车

房产

旅游

教育

时尚

科技

财经

娱乐

更多

登录狐友

0

分享到

的函数接口，除了待mock的接口。即被测类A和mock类B之间没有任何关系，mock类B中同样实现了被测类A中的大部分接口，在测试代码中，通过声明mock类B的对象，来达到测试目的。

上述两种方案都可以解决gmock不能mock非虚函数的问题，但是都并不完美，均有其缺点：方案一最大的问题是需要修改开发源码，这对于老工程来讲，几乎是不可能的，除非赶上开发重构代码；方案二虽然不会修改开发源码，但是需要维护一套开发代码，当开发代码有变更时，mock的类B需要进行同步修改，无疑加大了测试的维护成本。

如何解决？——Hook

提到hook，就不得不提百度在11年开源的Baiduhook，其提供了linux平台下C/C++程序的hook功能，可以解决gmock只能mock虚函数的限制。Linux上的hook和windows上的原理差不多，操作基本上是对目标函数进行劫持，替换成自己的函数，然后在自己的函数中进行一些用户预期的操作，比如修改函数返回值等。对hook原理比较感兴趣的可以拜读下源码：
<https://code.google.com/archive/p/baiduhook/>

看起来似乎可以解决我们的问题了，但是不幸的是，目前该hook技术仅支持了Linux平台，而我们的测试框架是在MAC OSX系统下搭建的，MAC OSX是Unix系统，bhook无法在MAC下使用。综合考虑后，决定在Linux系统进行导航引擎的单测。百度以及公司内部都基于hook以及gmock，对gtest进行了二次封装，形成了自己的单元测试框架btest和ttest。

(4) ttest和btest

24小时热文

1

最美“水
恋爱了

2

人工智
情内容

3

华硕Zei
820处理武警也
TFBOY郭明錤
解决供



新闻

体育

汽车

房产

旅游

教育

时尚

科技

财经

娱乐

更多

登录狐友

0

分享到

libbfd（二进制文件描述库）和libopcodes（程序调试，归档等）。

Øttest：须安装特定版本的binutils以及对应版本的gcc。

1) binutils版本不对

所有的case以及源码编译没有问题，但是在运行case的时候会出现如下图8所示的core：

```
(gdb) bt
#0 0x00007fc60b95e9b0 in ?? ()
#1 0x0000000000451d66 in AsmHelper::print_address(addr=4216766, info=0x7ffc5593b368) at AsmHelper.cpp:125
#2 0x00007fc60aeeaf48 in ?? () from /usr/lib/libopcodes-2.24-system.so
#3 0x00007fc60aeeef5d in ?? () from /usr/lib/libopcodes-2.24-system.so
#4 0x0000000000451e89 in AsmHelper::Copy(buf=0x1db1d00 "\270\207x\n\306\177", bufsize=44, abfd=0x1c5c330,
src=0x4057be <Test_C::method1(int)>, minsize=14, src_dis_len=0x7ffc5593b64c) at AsmHelper.cpp:52
#5 0x0000000000449379 in HookImpl::AttachFuncLib(this=0x1db1a70, symbol=...,
hook_func=0x40581a <tbase_Test_C_method1(Test_C*, int)>, pold_func=0x0) at HookImpl.cpp:65
#6 0x00000000004485af in HookManager::AttachFuncLibAll(this=0x1c5c080,
LibreOffice Calc  x684380 <g_tmock_tbase_func_str> "Test_C::method1(int)",
hook_func=0x40581a <tbase_Test_C_method1(Test_C*, int)>, pold_func=0x0) at HookManager.cpp:146
#7 0x0000000000447f51 in Core::AttachFuncLibAll(this=0x1c5c010,
func_name=0x684380 <g_tmock_tbase_func_str> "Test_C::method1(int)",
hook_func=0x40581a <tbase_Test_C_method1(Test_C*, int)>, pold_func=0x0) at ../core/Core.h:40
#8 0x0000000000447cfa in attach_func_lib_all(func_name=0x684380 <g_tmock_tbase_func_str> "Test_C::method1(int)",
hook_func=0x40581a <tbase_Test_C_method1(Test_C*, int)>) at tbase.cpp:96
#9 0x00000000004478d7 in attach_c_func_lib(func_with_param=0x684380 <g_tmock_tbase_func_str> "Test_C::method1(int)",
func_name=0x454a72 "Test_C::method1", hook_func=0x40581a <tbase_Test_C_method1(Test_C*, int)>)
at src/tbase_util.cpp:45
#10 0x00000000004076a2 in tmock_Test_C_method1::tmock_Test_C_method1(this=0x683f60 <Test_C_method1__mockobj>)
at tmock_class_method.cpp:33
#11 0x0000000000406742 in __static_initialization_and_destruction_0(__initialize_p=1, __priority=65535)
at tmock_class_method.cpp:33
#12 0x00000000004067dd in _GLOBAL_sub_I_tmock_class_method.cpp(void) () at tmock_class_method.cpp:73
#13 0x000000000045489d in __libc_csu_init()
#14 0x00007fc60a3ebcd5 in __libc_start_main(main=0x43b42a <main(int, char**)>, argc=1, argv=0x7ffc5593b098,
init=0x454850 <__libc_csu_init>, fini=<optimized out>, rtld_fini=<optimized out>) at libc-start.c:246
#15 0x00000000004056f9 in _start()
```

图8binutils版本错误引起的core

2) gcc版本不对

24小时热文

1

最美“水
恋爱了

2

人工智
情内容

3

华硕Zei
820处理



武警也
TFBOY



郭明錤
解决供



新闻

体育

汽车

房产

旅游

教育

时尚

科技

财经

娱乐

更多

登录狐友

0

分享到

reference to `testing::internal::FormatFileLocation`

Stack OverFlow上给的解释是：

I think you are victim of the ABI change introduced with libstdc++ in v5. They had to change the implementation of `std::string`, because C++11 mandates a particular implementation while was not the case before. That lead to changes in symbol names. The problem is specific to migrating from pre-v5 libstdc++ (gcc) to v5 or higher and should therefore not happen again.

Øbtest：仍需要特定版本的Linux系统以及gcc版本。

1) 虚拟机centOS4.3+gcc3.4.5

该虚拟机上安装的btest也只有相应的lib和so文件，没有btest的源码，直接运行自带的samples，btest运行完好，没有相应的core。

注：实际运行过程中对gdb版本也有要求（6.7及以上版本），否则会出现：

this=dwarf2_read_address: Corrupted DWARF expression。

2) 虚拟机centOS6.5

centOS4.3上整个测试框架运行没有问题，但是毕竟该版本的系统太老了，centOS官方已经停止维护了，各种软件包都没法通过yum来安装，这也给后续配置vim开发环境带来了一定程度的麻烦，所以，就想着能否用高版本的centOS来试下btest是否能运行，结果是不行的，同样会崩到系统库中。

24小时热文

1

最美“水
恋爱了

2

人工智
情内容

3

华硕Zei
820处理



武警也
TFBOY



郭明錤
解决供



新闻

体育

汽车

房产

旅游

教育

时尚

科技

财经

娱乐

更多

登录狐友

0

分享到

依赖于Linux系统的底层库，需要特定版本的系统库。虽然有了btest或者ttest，可以很方便的mock接口，但方便的同时，我们就不会再去思考如何对复杂接口进行解耦和了。

（5）有些函数扇出太高，可测性太低

有些历史接口，其扇出达到了40+，代码行也有900+，圈复杂度更是达到了400+，对这样的一类接口，几乎不具可测性，如果这类接口又是业务中很重要的接口，建议开发一起从可测性角度出发重新设计，达到可测性后再来开展单元测试。

五、UT和SDK测试的差异

（1）SDK测试的对象是公开的API，这些API有详细的接口说明文档。UT的测试对象是内部函数，这些函数没有任何文档，需要测试通过debug或者找开发咨询去了解。

（2）SDK测试可能只需要了解某个API被设计来干什么，对其内部如何设计关心的并不多。UT不单需要知道被测函数的功能是什么，还要了解其是如何设计的，实现原理是什么，要求比SDK测试要高。

（3）SDK测试除了要保证接口本身的功能外，更多的还要关心第三方使用者会如何用，即调用场景。UT不需要关心外部如何调，更加聚焦函数本身。

（4）数据构造，UT深入到函数内部，构造的数据不仅仅包含函数入参，还包含函数内部用到的一些数据。

24小时热文

1

最美“水
恋爱了

2

人工智
情内容

3

华硕Zei
820处理武警也
TFBOY郭明錤
解决供

0

分享到

护成本也很高。

将我们公众号置顶 不会漏掉我们的原创干货哦！ [返回搜狐，查看更多](#)

声明：本文由入驻搜狐号的作者撰写，除搜狐官方账号外，观点仅代表作者本人，不代表搜狐立场。

阅读 (1515)

不感兴趣

投诉



¥169



两件 ¥99元
每个细节都，让你怦然心动



¥149.00 ¥249

mate9
小米
小米5x
手机
s8

广告

我来说两句

0人参与，0条评论

来说两句吧.....

登录并发表

搜狐“我来说两句” 用户公约

还没有评论，快来抢沙发吧！

24小时热文

1

最美“水
恋爱了

2

人工智
情内容

3

华硕Zei
820处理



武警也
TFBOY



郭明錤
解决供

推荐

推荐阅读

- 苹果
- 人脸识别
- 新浪微博
- 饿了么
- 百度
- 比特币
- VR
- OPPO
- 优酷
- 百度外卖
- Snap



钛媒体 钛媒体 钛媒体APP · 今天 10:28

💬 3

别小看“搞笑诺贝尔奖”，要向好奇心致敬



虎嗅 虎嗅网 虎嗅网 · 今天 09:37

💬

支付宝之后，微信也终于放大招了！外媒：微信正在打造超级AR平台

太保乱谈 · 今天 10:09

💬

很多公司都能造"iPhone X"：为何还是苹果做出来了？

IT之家 · 今天 08:14

💬 2

全新奥迪A5轿跑家族。 现已格调上市

广告 · 今天 14:19

到底是什么在制约网络文学发展？

砍柴网_SUFFIX_MP · 今天 10:49

💬

24小时热文

- 1最美“水恋爱了”
- 2人工智能情内容
- 3华硕Zen 820处理
- 武警也TFBOY
- 郭明錤解决供



新闻

体育

汽车

房产

旅游

教育

时尚

科技

财经

娱乐

更多

登录狐友

和采坦崛起一便利店、办公全页采、日功告页机、元人零告



36氪 · 今天 08:59



凡客的荣辱兴衰，是一场雷军也救不回来的“中年危机”



爱范儿 · 今天 08:30

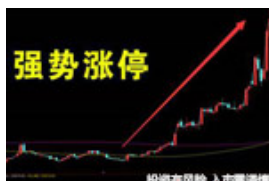


从新手村到史诗英雄，程序员的打怪升级之路

InfoQ · 今天 09:08



短线高手每天微信免费推荐3只短线牛股，可以观察几天看看效果，不收费



广告 · 今天 14:19

未来已无处不在，语音识别将改变的5个行业

24小时热文

1

最美“水
恋爱了

2

人工智
情内容

3

华硕Zei
820处理



武警也
TFBOY



郭明錤
解决供



新闻

体育

汽车

房产

旅游

教育

时尚

科技

财经

娱乐

更多

登录狐友



你觉得苹果在无线充电方面的意图是什么？



威锋网 · 今天 09:47



解密iPhone X大脑：苹果押注AI芯片，早在三年前就决定了



量子位 · 今天 12:54



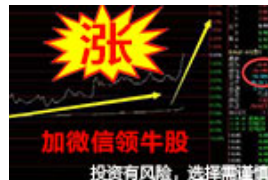
以幽默致敬科学，盘点那些你不知道的奇葩诺贝尔奖



极客视界 · 今天 13:58



这几只股打死你都不要卖，股民翻盘就靠它



广告 · 今天 14:19

由《小手牵小狗》聊爆款，优酷网综沙龙解密“易与难”

24小时热文

1

最美“水
恋爱了

2

人工智
情内容

3

华硕Zer
820处理



武警也
TFBOY



郭明錤
解决供



新闻

体育

汽车

房产

旅游

教育

时尚

科技

财经

娱乐

更多

登录狐友



砍柴网_SUFFIX_MP · 今天 13:54



今年的双 11 还没来，网购的快递纸箱就已经快不够用了



爱范儿 · 今天 13:26



双创周亚洲智能硬件大赛完美落幕，中国水下无人机项目脱颖而出



动点科技 · 今天 12:06



【破解DeepMind的游戏人工智能】AI现在只看2分钟视频就能创建游戏了



新智元 · 今天 13:06



汾酒集团出品，鸡年纪念酒震撼发售，赶紧抢两箱！



24小时热文

1

最美“水
恋爱了

2

人工智
情内容

3

华硕Zer
820处理



武警也
TFBOY




郭明錤
解决供


联通营业厅开始送快递，终于回归“祖业”？

 马继华 · 今天 11:10


💬




5个细微的改变，这将极大地改善聊天机器人

 太保乱谈 · 今天 10:35


💬




杨向华：过去七个月有6部网大在爱奇艺分账金额超千万

 砍柴网_SUFFIX_MP · 今天 13:22

💬



IROS 2017上，这些厂商将会给我们展示什么样的黑科技？
| IROS 2017

 雷锋网 · 今天 11:26

💬

中国股市打新股的方法彻底变了，现在知晓还不晚









广告 · 今天 14:19

加载更多

- 24小时热文
- 1

最美“水
恋爱了
- 2

人工智
情内容
- 3

华硕Zei
820处理
- 

武警也
TFBOY
- 

郭明錤
解决供