**下载gtest源码并解压**

$ git clone https://github.com/google/googletest

**CMake配置**

进入源目录，执行如下命令进行配置：

cmake CMakeLists.txt

**编译**执行如下命令，编译源码：

make

编译后会生成libgtest.a libgtest\_main.a两个静态库

**复制头文件到和生成的静态库你的测试项目目录**

头文件位置：

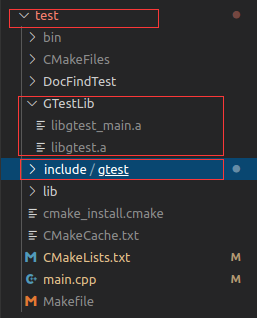
位于googletest/include/目录

静态文件位置：

位于lib目录

示例：

这里我将头文件放到include目录，静态库文件放到GTestLib目录，主目录test为我的测试项目目录



bin为测试程序生成目录

lib为测试库生成目录

**编写测试项目cmake文件**

cmake\_minimum\_required (VERSION 2.8)

project (DocFinderTest)

# 设置工作目录为上层目录

set(PROJECT\_SOURCE\_DIR ${PROJECT\_SOURCE\_DIR}/..)

# 添加 -pthread 选项

set(CMAKE\_CXX\_FLAGS "-std=c++11 -pthread ${CMAKE\_CXX\_FLAGS}")

# 指定编译所 include 搜索路径

include\_directories("include")

# 指定连接库的搜索目录

link\_directories(${PROJECT\_SOURCE\_DIR}/test/GTestLib)

add\_executable(test "main.cpp")

# 添加库依赖

target\_link\_libraries(test gtest gtest\_main)

# 指定可执行文件输出目录

set(EXECUTABLE\_OUTPUT\_PATH ${PROJECT\_SOURCE\_DIR}/test/bin)

注：编译选项需要指定-pthrea，不然会报错，原因不明

**简单的测试**

main.cpp如下：

#include<gtest/gtest.h>

#include<map>

#include<iostream>

using namespace std;

static int add(int l, int r)

{

    return l + r;

}

// 测试类

class TestMap:public testing::Test

{

public:

    // 第一个测试用例开始时调用

    static void SetUpTestCase()

    {

        cout<<"SetUpTestCase()"<<endl;

    }

    // 最后一个测试用例结束后调用

    static void TearDownTestCase()

    {

        cout<<"TearDownTestCase()"<<endl;

}

    // 每个测试用例开始时调用

    void SetUp()

    {

        cout<<"SetUp() is running"<<endl;

}

    // 每个测试用例结束后调用

    void TearDown()

    {

        cout<<"TearDown() is running"<<endl;

    }

};

// 测试用例，用例名为 Test1

TEST\_F(TestMap, Test1)

{

    // 断言，3+2 等于 5

    EXPECT\_EQ(add(3, 2), 5);

}

int main(int argc, char\*\* argv)

{

testing::InitGoogleTest(&argc, argv);

    return RUN\_ALL\_TESTS();

}

**编译并运行测试程序**

执行cmake命令

cmake CMakeLists.txt

执行make编译程序：

make

允许测试程序：

bin/test

输出结果如下：

[==========] Running 1 test from 1 test suite.

[----------] Global test environment set-up.

[----------] 1 test from TestMap

SetUpTestCase()

[ RUN ] TestMap.Test1

SetUp() is running

TearDown() is running

[ OK ] TestMap.Test1 (0 ms)

TearDownTestCase()

[----------] 1 test from TestMap (0 ms total)

[----------] Global test environment tear-down

[==========] 1 test from 1 test suite ran. (0 ms total)

[ PASSED ] 1 test.