



CMake的CTest方法

原创

2015年06月22日 17:28:35

1345



1

PART 1 :

参考：<http://hahack.com/codes/cmake/>

Demo目录结构如下：

```
[cpp]
1. Test/
2.   |— add.cpp
3.   |— CMakeLists.txt
```

add.cpp

```
[cpp]
1. #include <iostream>
2. #include <stdlib.h>
3. int main(int argc, char *argv[])
4. {
5.     if (argc != 3) {
6.         std::cout << "parameter error" << std::endl;
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！



种头发原理



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

🗣 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

他的最新文章

更多文章

登录

注册

```

10.     int a, b;
11.     a = atoi(argv[1]);
12.     b = atoi(argv[2]);
13.     std::cout << a << " + " << b << " is " << a + b << std::endl;
14.     return 0;
15. }

```

CMakeLists.txt



1

```

[cpp]
1.  CMAKE_MINIMUM_REQUIRED(VERSION 3.3)
2.  ADD_EXECUTABLE(add add.cpp)
3.  enable_testing()
4.  ADD_TEST(NAME test_add_2_3 COMMAND add 2 3)
5.  SET_TESTS_PROPERTIES(test_add_2_3
6.      PROPERTIES PASS_REGULAR_EXPRESSION "5")
7.  ADD_TEST(NAME test_add_4_5 COMMAND add 4 5)
8.  SET_TESTS_PROPERTIES(test_add_4_5
9.      PROPERTIES PASS_REGULAR_EXPRESSION "9")

```

在CMakeLists.txt里面，我们添加了两个测试用例。其中 PASS_REGULAR_EXPRESSION 用来测试输出是否包含后面的字符串。

在Test目录下建立build目录，cd build && cmake .., make, make test, 结果如下：

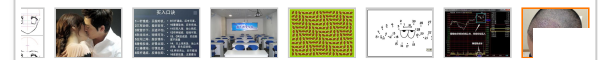
```

[cpp]
1.  huyanjie@debian:~/test/Test/build$ make test
2.  Running tests...
3.  Test project /home/huyanjie/test/Test/build
4.    Start 1: test_add_2_3
5.  1/2 Test #1: test_add_2_3 ..... Passed    0.00 sec
6.    Start 2: test_add_4_5
7.  2/2 Test #2: test_add_4_5 ..... Passed    0.00 sec

```



种头发原理



thrift 577

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

🗣 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

2015年8月

6篇

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

[登录](#)
[注册](#)

```
11. Total Test time (real) = 0.01 sec
```

可以看到两个用例都正确执行了。

像上面的方式写测试用例还是比较繁琐，还可以定义宏来简化：

```
[cpp]
1. CMAKE_MINIMUM_REQUIRED(VERSION 3.3)
2. ADD_EXECUTABLE(add add.cpp)
3. enable_test
4.
5. macro(do_test ARG1 ARG2 RESULT)
6.     ADD_TEST(test_add_${ARG1}_${ARG2} COMMAND add ${ARG1} ${ARG2})
7.     SET_TEST_PROPERTIES(test_add_${ARG1}_${ARG2}
8.         PROPERTIES PASS_REGULAR_EXPRESSION ${RESULT})
9. endmacro(do_test)
10. do_test(2 3)
11. do_test(4 5 9)
```

PART 2:

CPPUNIT使用如下：

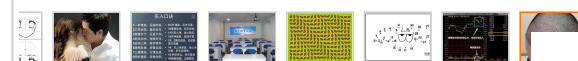
参考<http://blog.csdn.net/freemalcom/article/details/753819>

```
[cpp]
1. #include <cppunit/TestFixture.h>
2. #include <cppunit/TestRunner.h>
3. #include <cppunit/TestResult.h>
4. #include <cppunit/TestRunner.h>
5. #include <cppunit/extensions/HelperMacros.h>
6.
7. class StringTest : public CppUnit::TestFixture
8. {
9.     CPPUNIT_TEST_SUITE(StringTest);
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！



种头发原理



thrift Processor

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

🗣 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

登录

注册

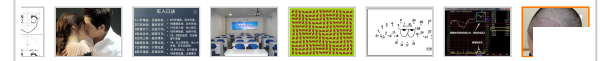
```

13. public:
14.     void setUp()
15.     {
16.         m_str1 = "Hello, world";
17.         m_str2 = "Hi, cppunit";
18.     }
19.     void tearDown()
20.     {
21.     }
22.     void testSwap()
23.     {
24.         std::string str1 = m_str1;
25.         std::string str2 = m_str2;
26.         m_str1.swap(m_str2);
27.         CPPUNIT_ASSERT(m_str1 == str2);
28.         CPPUNIT_ASSERT(m_str2 == str2);
29.     }
30.     void testFind()
31.     {
32.         int pos1 = m_str1.find(',');
33.         int pos2 = m_str2.rfind(',');
34.         CPPUNIT_ASSERT_EQUAL(5, pos1);
35.         CPPUNIT_ASSERT_EQUAL(2, pos2);
36.     }
37.
38. protected:
39.     std::string m_str1;
40.     std::string m_str2;
41. };
42.
43. CPPUNIT_TEST_SUITE_REGISTRATION(StringTest);
44. int main(int argc, char *argv[])
45. {
46.     CppUnit::TestResult r;
47.     CppUnit::TestResultCollector rc;
48.     r.addListener(&rc);
49.

```



种头发原理



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

🗣 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

```
53.
54.     CppUnit::TextOutputter o(&rc, std::cout);
55.     o.write();
56.
57.     return rc.wasSuccessful()?0:-1;
58. }
```

PART 3

CMake 结合CPPUnit



1



版权声明：本文为博主原创文章，未经博主允许不得转载。

<http://blog.csdn.net/huyanjie0327/article/details/46652313>



目前您尚未登录，请 [登录](#) 或 [注册](#) 后进行评论

cmake 使用 ctest 测试



EastonWoo 2015年09月12日 13:06 4015

cmake 使用 ctest 测试 1). test.cpp /// @file test.cpp /// @brief /// @author EastonWoo //...

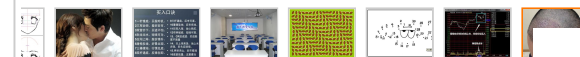
加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册



种头发原理



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

🗣 QQ客服 🗣 客服论坛

[关于](#) [招聘](#) [广告服务](#) [百度](#)

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

希望这是现阶段阻碍阅读shiboken和PySide源码的涉及cmake的最后一个障碍 ^_^ 学习 cmake 的单元测试部分 ctest。简单使用 最简单的使用ctest的方法，就是...

程序猿不会英语怎么行？英语文档都看不懂！

软件工程出身的英语渣教你用数学公式读懂天下英文 →



1

CMake创建CTest工程



sheismylife

2011年10月04日 10:35 7381

在CMake工程顶层目录内的CMakeLists.txt文件中添加最后一行，使得目录结构中包含test目录，并且编译完成时会在build目录下产生test_bin cmake_minimum_re...

CMake中文手册

2015年12月14日 11:11 856KB

下载



CMake 7 步教程



shutdown_r_now

2016年04月09日 10:25 1120

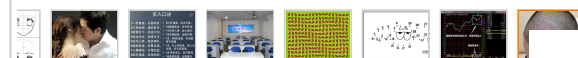
本文译自官方简明教程，但有所删改。本文中所涉及到的所有源码，可以从 CMake 源码包的 Tests/Tutorial 目录中取得。题外话：在本文翻译到差不多一半的时候，我很悲剧地发现，已经有别人...

程序员不会英语怎么行？

北大猛男教你：不背单词和语法，一个公式学好英语



种头发原理



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

💬 QQ客服 🗨 客服论坛

[关于](#) [招聘](#) [广告服务](#) [百度](#)

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

[登录](#)[注册](#)

CMake是一个比make更高级的编译配置工具，它可以根据不同平台、不同的编译器，生成相应的Makefile或者vcproj项目。通过编写CMakeLists.txt，可以控制生成的Makefile...

使用CMAKE编译单元测试代码发现如下问题



Dalek

2012年02月09日 11:54

643

Linking CXX executable xmltest [84%] Built target xmltest Scanning dependencies of target listent...



c++单元测试框架Gtest的简单配置与使用



clayandwind

2015年09月20日 21:11

1363

之前从来没有写测试代码，在代码调试上确实花了很多时间。最近在看《Agile Java》，逐渐了解到单元测试在软件开发过程中的重要性，觉得应该从现在开始养成写单元测试的习惯。Java有大名鼎鼎的...



<c:if test>标签使用



yunpyy

2013年09月12日 19:09

10019

1>test命令之数字test 语法：[number relation number] 通过关系运算符来对数字进行比较 关系运算符： -lt 小于 -le 小于...

ctest

2011年11月27日 12:43 433KB

[下载](#)

CMake的CTest方法



huyanjie0327

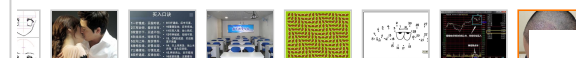
2015年06月26日 17:28

1345

参考：http://hahack.com/codes/cmake/ Demo目录结构如下：Test/ |—— add.cpp |—— CMakeLists.txt add.cpp #incl u...



种头发原理



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

🗣 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

怎样快速恢复视力

怎么恢复视力

百度广告

CTest



Bruce_0712 2016年12月14日 14:50 217

一、初识CTest CTest是CMake集成的一个测试工具，在使用CMakeLists.txt文件编译工程的时候，CTest会自动configure、build、test和展现测试结果...



c++总结（一）



u012579879 2013年11月06日 12:11 774

1，若char是一字节，int是4字节，指针类型是4字节，代码如下：Class CTest { Public: CTest():m_chData('\0'),m_nData(0) { ...

CMake教程及例子



houqiqi 2015年05月09日 20:17 2127

转 自： LaineGates的专栏， <http://blog.csdn.net/lainegates> CMake 简明教程（1）---第一个例子 原文网址：<http://cmake...>

值得推荐的C/C++框架和库 (真的很强大)



Vincent95 2017年05月15日 16:33 1708

转自：<http://blog.csdn.net/xiaoxiaoyeyaya/article/details/42541419>（侵权）目录(?)[+] 值得学习的C语言开源项目 - ...

最全面的 C++ 资源、框架大全

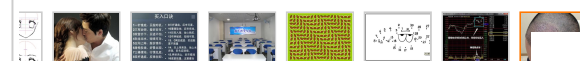


WAN_EXE 2017年04月13日 19:13 492

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！



种头发原理



联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

🗣 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

福利！免费试听深度学习！

人工智能必备，迈出百万年薪第一步！



MyMFC(7-9)对话 CTestDlg



u010111422

2014年09月12日 16:48

341

// TestDlg.cpp : 实现文件
#include "stdafx.h" #include "MyMFC2-Dialog.h" #include "TestDlg.h" #inclu...



CMake 添加头文件目录，链接动态、静态库（添加子文件夹）



CMake支持大写、小写、混合大小写的命令。cmake 添加头文件目录，链接动态、静态库1. 添加头文件目录INCLUDE_DIRECTORIES它相当于 g++ 选项中的 -I 参数的作用，也相当...



lanchunhui

2017年02月26日 23:19

2177

CWinThread类的封装方法



limingmcu

2014年08月14日 12:32

493

.h 文件 #define WM_TEST WM_USER + 1 class CTestThread : public CWinThread { DECLARE_DYNCREATE(C...

使用Setup Factory 制作安装包的一般过程



diyuzhizhu

2016年09月13日 16:02

1216

安装包是用户接触到工程的第一步，所以安装包的制作非常重要。下面记录一下使用Setup Factory制作安装包的过程。建立工程打开setupFactory，按照提示一步步往下走即可。在下图...



种头发原理



联系我们



请扫描二维码联系客服

webmaster@csdn.net

400-660-0108

QQ客服 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册