

GoogleTest快速入门

[原文链接](#)

用 GoogleTest 测试 C++代码

如果要测试这样的代码：

```
1 // calc.hpp
2 #pragma once
3
4 class Calc {
5 public:
6     int add(int a, int b);
7 };
```

```
1 // calc.cpp
2 #include "calc.hpp"
3
4 int Calc::add(int a, int b) {
5     return a + b;
6 };
```

单元测试就要这样写：

```
1 // calc_test.cpp
2 #include <gtest/gtest.h>
3 #include "calc.hpp"
4
5 TEST(calc, add) {
6     ASSERT_EQ(7, Calc().add(3, 4));
7     ASSERT_EQ(150, Calc().add(50, 100));
8 }
```

然后编译连接

```
1 # g++ -c calc.cpp -o calc.o
2 # g++ -c calc_test.cpp -o calc_test.o
3 # g++ calc.o calc_test.o -lgtest -lgtest_main -pthread -o thetest
```

执行

1
2
3
4
5
6
7
8
9
1
0
1
1
1
2

```
# ./thetest
Running main() from /build/googletest-j5yxiC/googletest-1.10.0/googletest/src/gtest_main.cc
[=====] Running 1 test from 1 test suite.
[-----] Global test environment set-up.
[-----] 1 test from calc
[ RUN      ] calc.add
[          OK ] calc.add (0 ms)
[-----] 1 test from calc (0 ms total)

[-----] Global test environment tear-down
[=====] 1 test from 1 test suite ran. (0 ms total)
[ PASSED  ] 1 test.
```

用 GoogleTest 测试 C 代码

由于 GoogleTest 本身是 C++ 的，要测试 C 代码就要使用 `extern "C"`。

如果要测试这样的代码：

```
1 // calc.h
2 #pragma once
3
4 int add(int a, int b);
```

```
1 // calc.c
2 #include "calc.h"
3
4 int add(int a, int b) {
5     return a + b;
6 }
```

这时需要在 `calc_test.cpp` 里，用 `extern "C"` 包裹 `#include "calc.h"`。

```

1 // calc_test.cpp
2 #include <gtest/gtest.h>
3 extern "C" { // 在这里包裹
4 #include "calc.h"
5 }
6
7 TEST(calc, add) {
8     ASSERT_EQ(7, add(3, 4));
9     ASSERT_EQ(150, add(50, 100));
10 }

```

编译连接：

```

1 # gcc -c calc.c -o calc.o
2 # g++ -c calc_test.cpp -o calc_test.o
3 # g++ calc.o calc_test.o -lgtest -lgtest_main -pthread -o thetest

```

执行：

```

1
2
3 # ./thetest
4 Running main() from /build/googletest-j5yxiC/googletest-1.10.0/googletest/src/gtest_main.cc
5 [=====] Running 1 test from 1 test suite.
6 [-----] Global test environment set-up.
7 [-----] 1 test from calc
8 [ RUN      ] calc.add
9 [          OK ] calc.add (0 ms)
10 [-----] 1 test from calc (0 ms total)
11
12 [-----] Global test environment tear-down
13 [=====] 1 test from 1 test suite ran. (0 ms total)
14 [  PASSED  ] 1 test.
15
16
17
18
19
20

```