

```
1 // Example code uses DEADLOCK.
2 // programmer: supakit@it.kmitl.ac.th
3 // 2017 Mar 30 completed
4
5 #include <stdio.h>
6 #include <thread>
7 #include <mutex>
8
9
10 #define NUM_THREADS 2
11
12 static int threadData[NUM_THREADS];
13 int g;
14 std::mutex g_mutex1, g_mutex2;
15
16
17
18 void func_mycounter1(void) {
19     int id = 1;
20     std::unique_lock<std::mutex> mutex1(g_mutex1, std::defer_lock);
21     std::unique_lock<std::mutex> mutex2(g_mutex2, std::defer_lock);
22
23     printf("\n thread[%d]--start", id);
24     mutex1.lock();
25     for (int i = 0; i < 1e6; i++)
26         g = g + 1;
27     printf("\n wait to do(1)");
28     mutex2.lock();
29     for (int i = 0; i < 1e6; i++)
30         g = g + 1;
31     printf("\n [%d] running", id);
32     printf("\n\t\t\t thread[%d]--finish", id);
33     mutex2.unlock();
34     mutex1.unlock();
35
36 }
37
38
39 void func_mycounter2(void) {
40     int id = 2;
41     std::unique_lock<std::mutex> mutex1(g_mutex1, std::defer_lock);
42     std::unique_lock<std::mutex> mutex2(g_mutex2, std::defer_lock);
43
44     printf("\n thread[%d]--start", id);
45     mutex2.lock();
46     for (int i = 0; i < 1e6; i++)
47         g = g + 1;
48     printf("\n wait to do(2)");
49     mutex1.lock();
50     for (int i = 0; i < 1e6; i++)
51         g = g + 1;
52     printf("\n [%d] running", id);
53     printf("\n\t\t\t thread[%d]--finish", id);
54     mutex1.unlock();
55     mutex2.unlock();
56 }
```

```
57
58 int main(void) {
59     std::thread threadA;
60     std::thread threadB;
61
62     g = 0;
63     printf("\n start g = %d", g);
64
65     threadA = std::thread(func_mycounter1);
66     threadB = std::thread(func_mycounter2);
67
68
69     threadA.join();
70     threadB.join();
71
72     printf("\n finish g = %d", g);
73     getchar();
74     return 0;
75 }
```