

```
1 // Example code uses CRITICAL SECTION & MUTUAL EXCLUSION.
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 5
11
12 static int threadData[NUM_THREADS];
13 int g;
14 std::mutex g_mutex;
15
16
17 void func_mycounter(int id) {
18     std::unique_lock<std::mutex> chkCritical(g_mutex, std::defer_lock);
19     int i;
20     printf("\n thread[%d]--start", id);
21     chkCritical.lock();
22     printf("\n [%d] running", id);
23     for (int i=0; i < 1e6; i++){
24         g = g + 1 ;
25     }
26     chkCritical.unlock();
27     printf("\n\t\t\t thread[%d]--finish", id);
28 }
29
30 int main(void) {
31     std::thread threadId[NUM_THREADS];
32     printf("\n-----MULTITHREAD-----");
33     g = 0;
34     printf("\n start g = %d", g);
35
36     for (int i = 0; i < NUM_THREADS; i++) {
37         threadId[i] = std::thread(func_mycounter, i);
38     }
39
40     for (int i = 0; i < NUM_THREADS; i++){
41         threadId[i].join();
42     }
43
44     printf("\n finish g = %d", g);
45
46     /*
47     printf("\n-----SINGLE THREAD-----");
48     g = 0;
49     printf("\n start g = %d", g);
50     for (int i = 0; i < 5; i++) {
51         func_mycounter(i);
52     }
53     printf("\n finish g = %d", g);
54     */
55     getchar();
56     return 0;
}
```

57 }