

Synchronization: Two-Process Solutions, MUTEX, and Semaphore

Task#1

Solve CS problem using semaphore.

```
1#include <iostream>
2#include <stdlib.h>
3#include <stdio.h>
4#include <pthread.h>
5#include <semaphore.h>
6#include <unistd.h>
7#include <thread>
8
9using namespace std;
10
11sem_t lk;
12int counter = 0;
13
14void* thread_1(void* args){
15    // CS1
16    // Semaphore Lock
17    sem_wait(&lk); // Wait
18    for(int i = 0; i <= 5000000; i++){
19        counter++;
20        if(counter % 1000000 == 0){
21            cout << "Val of counter => TH -> 1, 1" << counter << endl;
22        }
23    }
24
25    sem_post(&lk); // unlock
26
27
```

```
28    // ----- //
29    sem_wait(&lk); // Wait
30    for(int i = 0; i <= 5000000; i++){
31        counter++;
32        if(counter % 1000000 == 0){
33            cout << "Val of counter => TH -> 1, 2" << counter << endl;
34        }
35    }
36
37    sem_post(&lk); // Unlock
38
39    // ----- //
40    sem_wait(&lk); // Wait
41    for(int i = 0; i <= 5000000; i++){
42        counter++;
43        if(counter % 1000000 == 0){
44            cout << "Val of counter => TH -> 1, 3" << counter << endl;
45        }
46    }
47
48    sem_post(&lk); // Wait
49
50    return NULL;
51}
52
```

```

53 void* thread_2(void* args){
54     sem_wait(&lk); // wait
55     for(int i = 0; i <= 5000000; i++){
56         counter--;
57         if(counter % 1000000 == 0){
58             cout << "Val of counter => TH -> 2" << counter << endl;
59         }
60     }
61     cout << "T2 example" << endl;
62
63     // Semaphore unlock
64     sem_post(&lk);
65
66     return NULL;
67 }
68
69 int main(){
70     sem_init(&lk, 0, 1);
71     pthread_t t1, t2;
72
73     pthread_create(&t1, NULL, thread_1, NULL);
74     pthread_create(&t2, NULL, thread_2, NULL);
75
76     pthread_join(t1, NULL);
77     pthread_join(t2, NULL);
78
79     cout << "Fin val -> Counter " << counter << endl;
80     return 0;
81 }

```

```
sam@sam-VirtualBox:~/Desktop$ gedit lab10.cpp
sam@sam-VirtualBox:~/Desktop$ g++ lab10.cpp -o lbs -pthread
sam@sam-VirtualBox:~/Desktop$ gedit lab10.cpp
sam@sam-VirtualBox:~/Desktop$ ./lbs
Val of counter => TH -> 1, 11000000
Val of counter => TH -> 1, 12000000
Val of counter => TH -> 1, 13000000
Val of counter => TH -> 1, 14000000
Val of counter => TH -> 1, 15000000
Val of counter => TH -> 25000000
Val of counter => TH -> 24000000
Val of counter => TH -> 23000000
Val of counter => TH -> 22000000
Val of counter => TH -> 21000000
Val of counter => TH -> 20
T2 example
Val of counter => TH -> 1, 21000000
Val of counter => TH -> 1, 22000000
Val of counter => TH -> 1, 23000000
Val of counter => TH -> 1, 24000000
Val of counter => TH -> 1, 25000000
Val of counter => TH -> 1, 36000000
Val of counter => TH -> 1, 37000000
Val of counter => TH -> 1, 38000000
Val of counter => TH -> 1, 39000000
Val of counter => TH -> 1, 310000000
Fin val -> Counter 10000002
sam@sam-VirtualBox:~/Desktop$
```

Task#2

Solve CS problem using MUTEX.

```

1 #include <iostream>
2 #include <pthread.h>
3 #include <thread>
4 #include <mutex>
5 using namespace std;
6
7 int counter = 0;
8 pthread_mutex_t lk;
9
10 std::mutex m1, m2;
11
12 void* thread_2(void *args){
13     pthread_mutex_lock(&lk);
14     for(int i = 0; i <= 5000000; i++){
15         counter--;
16         if(counter % 1000000 == 0){
17             cout << "cnt -> T2" << counter << endl;
18         }
19     }
20     pthread_mutex_unlock(&lk);
21
22     return NULL;
23 }
24
25 void* thread_1(void* args){
26
27     // CS1

```

```

28     // Semaphore Lock
29     pthread_mutex_lock(&lk);
30     for(int i = 0; i <= 5000000; i++){
31         counter++;
32         if(counter % 1000000 == 0){
33             cout << "Val of counter => TH -> 1, 1" << counter << endl;
34         }
35     }
36
37     pthread_mutex_unlock(&lk);
38
39     // ----- //
40     pthread_mutex_lock(&lk);
41     for(int i = 0; i <= 5000000; i++){
42         counter++;
43         if(counter % 1000000 == 0){
44             cout << "Val of counter => TH -> 1, 2" << counter << endl;
45         }
46     }
47
48     pthread_mutex_unlock(&lk);
49
50     // ----- //
51     pthread_mutex_lock(&lk);
52     for(int i = 0; i <= 5000000; i++){
53         counter++;
54         if(counter % 1000000 == 0){

```

```

52     for(int i = 0; i <= 5000000; i++){
53         counter++;
54         if(counter % 1000000 == 0){
55             cout << "Val of counter => TH -> 1, 3" << counter << endl;
56         }
57     }
58     pthread_mutex_unlock(&lk);
59
60     return NULL;
61 }
62
63 int main(){
64     pthread_mutex_init(&lk, NULL);
65     pthread_t t1, t2;
66
67     pthread_create(&t1, NULL, thread_1, NULL);
68     pthread_create(&t2, NULL, thread_2, NULL);
69
70     pthread_join(t1, NULL);
71     pthread_join(t2, NULL);
72
73     cout << "Fin val -> Counter " << counter << endl;
74     return 0;
75 }
76
77

```

```

sam@sam-VirtualBox:~/Desktop$ gedit lab10M.cpp
sam@sam-VirtualBox:~/Desktop$ gedit lab10M.cpp
sam@sam-VirtualBox:~/Desktop$ g++ lab10M.cpp -o lb -pthread
sam@sam-VirtualBox:~/Desktop$ ./lb
cnt -> T2-1000000
cnt -> T2-2000000
cnt -> T2-3000000
cnt -> T2-4000000
cnt -> T2-5000000
Val of counter => TH -> 1, 1-5000000
Val of counter => TH -> 1, 1-4000000
Val of counter => TH -> 1, 1-3000000
Val of counter => TH -> 1, 1-2000000
Val of counter => TH -> 1, 1-1000000
Val of counter => TH -> 1, 10
Val of counter => TH -> 1, 21000000
Val of counter => TH -> 1, 22000000
Val of counter => TH -> 1, 23000000
Val of counter => TH -> 1, 24000000
Val of counter => TH -> 1, 25000000
Val of counter => TH -> 1, 36000000
Val of counter => TH -> 1, 37000000
Val of counter => TH -> 1, 38000000
Val of counter => TH -> 1, 39000000
Val of counter => TH -> 1, 310000000
Fin val -> Counter 10000002
sam@sam-VirtualBox:~/Desktop$

```