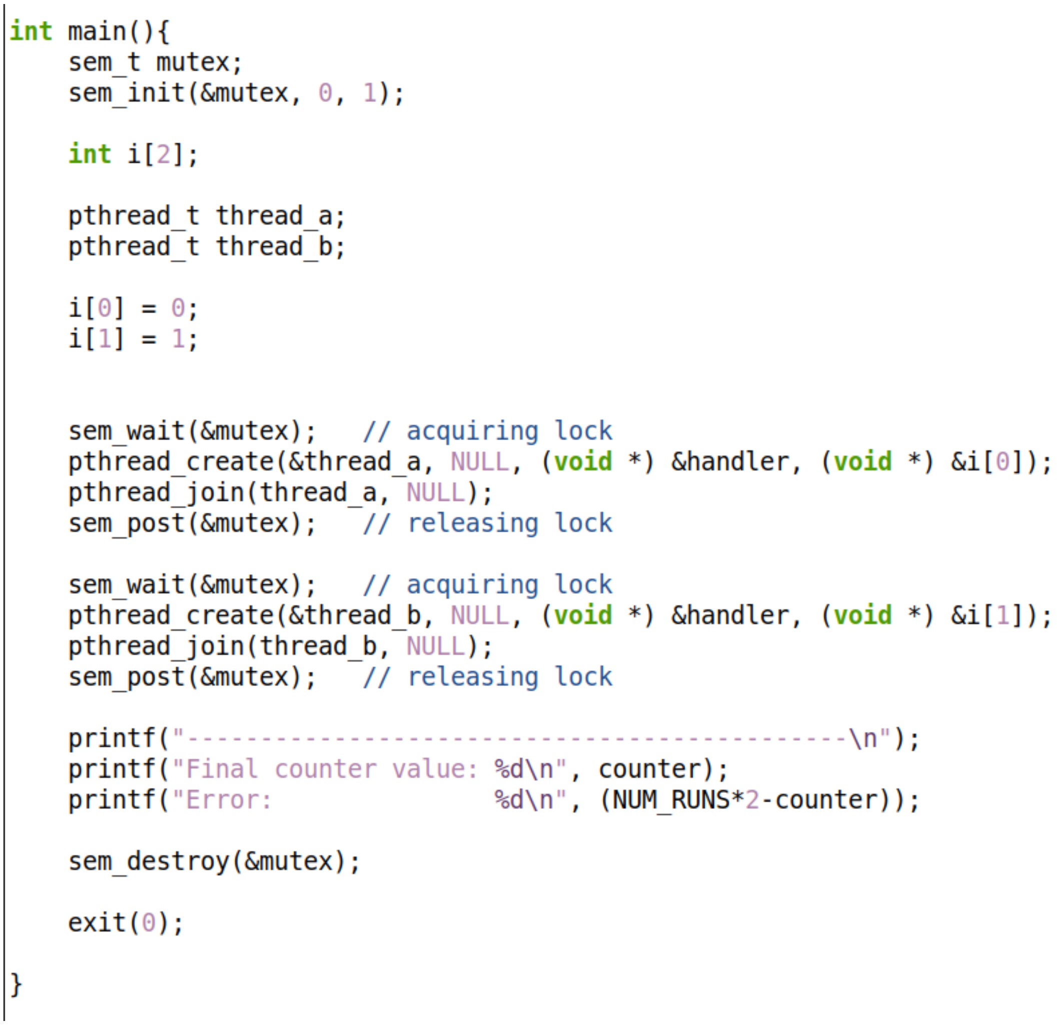
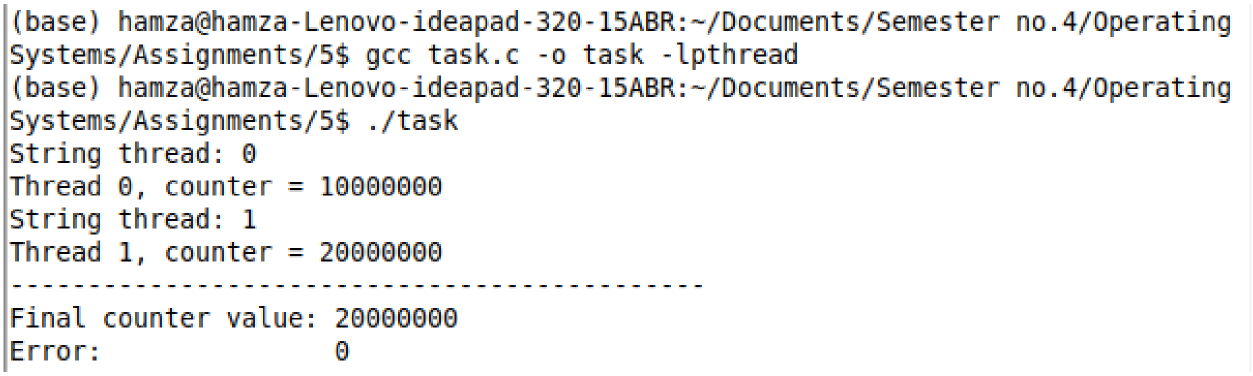
Operating Systems
Assignment # 05

Hamza Shahid
BSSE-4A
P200117



```c
int main(){
    sem_t mutex;
    sem_init(&mutex, 0, 1);

    int i[2];

    pthread_t thread_a;
    pthread_t thread_b;

    i[0] = 0;
    i[1] = 1;


    sem_wait(&mutex);   // acquiring lock
    pthread_create(&thread_a, NULL, (void *) &handler, (void *) &i[0]);
    pthread_join(thread_a, NULL);
    sem_post(&mutex);   // releasing lock

    sem_wait(&mutex);   // acquiring lock
    pthread_create(&thread_b, NULL, (void *) &handler, (void *) &i[1]);
    pthread_join(thread_b, NULL);
    sem_post(&mutex);   // releasing lock

    printf("---------------------------------------------\n");
    printf("Final counter value: %d\n", counter);
    printf("Error:               %d\n", (NUM_RUNS*2-counter));

    sem_destroy(&mutex);

    exit(0);

}
```



```
(base) hamza@hamza-Lenovo-ideapad-320-15ABR:~/Documents/Semester no.4/Operating
Systems/Assignments/5$ gcc task.c -o task -lpthread
(base) hamza@hamza-Lenovo-ideapad-320-15ABR:~/Documents/Semester no.4/Operating
Systems/Assignments/5$ ./task
String thread: 0
Thread 0, counter = 10000000
String thread: 1
Thread 1, counter = 20000000
---------------------------------------------
Final counter value: 20000000
Error:               0
```

The final counter value should be: 20000000
Now it is correct and the error is zero.

So, we have to acquire lock while creating and joining threads.