

main.c



Share

Run

Output

```
1 #include <stdio.h>
2 #include <pthread.h>
3 #define NUM_THREADS 5
4 pthread_mutex_t lock;
5 void* threadFunction(void* arg) {
6     pthread_mutex_lock(&lock);
7     printf("Thread %d is in the critical section.\n", *(int*)arg);
8     pthread_mutex_unlock(&lock);
9     return NULL;
10 }
11 int main() {
12     pthread_t threads[NUM_THREADS];
13     int threadIds[NUM_THREADS];
14     pthread_mutex_init(&lock, NULL);
15     for (int i = 0; i < NUM_THREADS; i++) {
16         threadIds[i] = i;
17         pthread_create(&threads[i], NULL, threadFunction, &threadIds[i]);
18     }
19     for (int i = 0; i < NUM_THREADS; i++) {
20         pthread_join(threads[i], NULL);
21     }
22     pthread_mutex_destroy(&lock);
23     return 0;
24 }
25
```

Thread 0 is in the critical section.
Thread 2 is in the critical section.
Thread 1 is in the critical section.
Thread 3 is in the critical section.
Thread 4 is in the critical section.

=== Code Execution Successful ===

main.c



Share

Run

Output

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <pthread.h>
4 #include <semaphore.h>
5 #include <unistd.h> // for sleep function
6
7 #define READERS 5
8 #define WRITERS 2
9
10 sem_t mutex, writeBlock;
11 int readCount = 0;
12
13 void* reader(void* arg) {
14     int f = *((int*)arg);
15     while (1) {
16         sem_wait(&mutex); // Enter critical section
17         readCount++;
18         if (readCount == 1) {
19             sem_wait(&writeBlock); // First reader blocks writers
20         }
21         sem_post(&mutex); // Exit critical section
22
23         // Reading section
24         printf("Reader %d is reading\n", f);
25         sleep(1); // Simulate reading time
26
27         sem_wait(&mutex); // Enter critical section
```

```
Reader 1 is reading
Reader 3 is reading
Reader 4 is reading
Reader 5 is reading
Reader 2 is reading
Writer 1 is writing
Writer 2 is writing
Reader 1 is reading
Reader 3 is reading
Reader 4 is reading
Reader 2 is reading
Reader 5 is reading
Writer 1 is writing
Writer 2 is writing
Reader 1 is reading
Reader 3 is reading
Reader 4 is reading
Reader 2 is reading
Reader 5 is reading
Writer 1 is writing
```