#include <stdio.h>

#include <pthread.h>

#include <semaphore.h>

#include <unistd.h>

#define NUM\_READERS 5

#define NUM\_WRITERS 2

int shared\_data = 0;

sem\_t mutex, writeblock;

int reader\_count = 0;

void \*reader(void \*arg) {

while (1) {

usleep(100000); // Sleep for a short time to simulate reading

sem\_wait(&mutex);

reader\_count++;

if (reader\_count == 1) {

sem\_wait(&writeblock);

}

sem\_post(&mutex);

// Reading the shared data

printf("Reader is reading: %d\n", shared\_data);

sem\_wait(&mutex);

reader\_count--;

if (reader\_count == 0) {

sem\_post(&writeblock);

}

sem\_post(&mutex);

}

}

void \*writer(void \*arg) {

while (1) {

sem\_wait(&writeblock);

// Modifying the shared data

shared\_data++;

printf("Writer is writing: %d\n", shared\_data);

sem\_post(&writeblock);

usleep(100000); // Sleep for a short time to simulate writing

}

}

int main() {

pthread\_t reader\_threads[NUM\_READERS];

pthread\_t writer\_threads[NUM\_WRITERS];

sem\_init(&mutex, 0, 1);

sem\_init(&writeblock, 0, 1);

for (int i = 0; i < NUM\_READERS; i++) {

pthread\_create(&reader\_threads[i], NULL, reader, NULL);

}

for (int i = 0; i < NUM\_WRITERS; i++) {

pthread\_create(&writer\_threads[i], NULL, writer, NULL);

}

for (int i = 0; i < NUM\_READERS; i++) {

pthread\_join(reader\_threads[i], NULL);

}

for (int i = 0; i < NUM\_WRITERS; i++) {

pthread\_join(writer\_threads[i], NULL);

}

sem\_destroy(&mutex);

sem\_destroy(&writeblock);

return 0;

}