```
#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

sem_t rw_mutex, mutex, queue;
int read_count = 0, data = 0;
void *reader(void *arg) {
int id = *(int*)arg;
sem_wait(&queue);
sem_wait(&mutex);
read_count++;

if (read_count == 1) sem_wait(&rw_mutex);
sem_post(&mutex);
sem_post(&queue);
printf("Reader %d reads %d\n", id, data);
sleep(1);
sem_wait(&mutex);
read_count--;
if (read_count == 0) sem_post(&rw_mutex);
sem_post(&mutex);
return NULL;
}

void *writer(void *arg) {
int id = *(int*)arg;
sem_wait(&queue);
sem_wait(&rw_mutex);
data++;
printf("Writer %d writes %d\n", id, data);
sleep(1);
sem_post(&rw_mutex);
sem_post(&queue);
return NULL;
}

int main() {
pthread_t r[3], w[2];
int i, id[5];
sem_init(&rw_mutex, 0, 1);
sem_init(&mutex, 0, 1);
sem_init(&queue, 0, 1);
for (i = 0; i < 3; i++) {
id[i] = i+1;
pthread_create(&r[i], NULL, reader, &id[i]);
}

for (i = 0; i < 2; i++) {
id[i+3] = i+1;
pthread_create(&w[i], NULL, writer, &id[i+3]);
}
```

```
for (i = 0; i < 3; i++) pthread_join(r[i], NULL);
for (i = 0; i < 2; i++) pthread_join(w[i], NULL);
sem_destroy(&rw_mutex);
sem_destroy(&mutex);
sem_destroy(&queue);
return 0;
}
```